AD-A054 475 WHITE SANDS MISSILE RANGE N MEX INSTRUMENTATION DIRE--ETC F/G 17/8
CONCEPTS IN REAL-TIME VIDEO TRACKING.(U)
MAY 78 A L GILBERT, M K GILES

UNCLASSIFIED STEWS-ID-78-1 NL

END
DATE FILMED
6 -78
DDC

TECHNICAL REPORT

(12)

B.S.

STEWS - ID - 78 - 1

CONCEPTS IN REAL-TIME VIDEO TRACKING

MAY 1978

FINAL REPORT

INSTRUMENTATION DIRECTORATE
US ARMY WHITE SANDS MISSILE RANGE
WHITE SANDS MISSILE RANGE, NEW MEXICO 88002

Destroy this report when no longer needed. Do not return it to the originator.

Disclaimer

The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: STEWS-ID-78-1
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): CONCEPTS IN REAL-TIME VIDEO TRACKING
5. TYPE OF REPORT & PERIOD COVERED: Final Report, Sep 77 - Mar 78
6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): A. L. Gilbert and M. K. Giles
8. CONTRACT OR GRANT NUMBER(s): 13 p3
9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Commander
US Army White Sands Missile Range
ATTN: STEWS-ID-T
White Sands Missile Range, New Mexico 88002
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
WSMR Task: IT008F-H
DA Project Number:
IT665804DE93-03
11. CONTROLLING OFFICE NAME AND ADDRESS
12. REPORT DATE: May 1978
13. NUMBER OF PAGES: 89
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 94p.
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Video Tracking
Real Time
Digital Image Processing
Pattern Recognition

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the real-time video (RTV) tracking system which will be deployed at White Sands Missile Range (WSMR) during fiscal year 1979. This tracking system utilizes a distributive array of five high-speed microprocessors to implement a real-time tracking algorithm with sufficient intelligence to overcome the tracking problems associated with typical noisy and cluttered backgrounds encountered in WSMR tracking imagery. Target positions relative to boresight and target attitude angles are also computed and recorded for each video field during the tracking sequence, thus eliminating the cost and delay

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

184 230

Block 20. ABSTRACT (continued)

imposed by present systems which require post-flight processing of film to extract these parameters.

A computer simulation of the RTV tracking system has been developed at WSMR to test the tracking and image processing algorithms utilized in the system. The simulation program is described, along with other research tools developed as part of the WSMR image processing laboratory. Finally, significant results of an investigation of various image processing techniques suitable for RTV tracking are presented.

PREFACE

This report presents the results of image processing research conducted at White Sands Missile Range (WSMR) during the period September 1977 through March 1978. A concise description of the final design of the real-time video tracking system presently being assembled at New Mexico State University is also included.

Principal investigators contributing to this image processing research are Dr. M. K. Giles and Dr. A. L. Gilbert. Dr. J. M. Taylor of NMSU helped with the simulation program, and Dr. R. Machuca of NMSU provided expert programming assistance. Mr. A. Garcia of WSMR also helped in the operation, documentation, and maintenance of the image processing laboratory. Mrs. R. Granger of WSMR prepared the final report manuscript with help from Mr. M. Ramos in preparing the figures.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (continued)

INTRODUCTION

A variety of methods of image data processing have become known over the past decade. Much of the effort has been sponsored by the Defense Advanced Research Projects Agency (DARPA) and by NASA to further scientific understanding of imagery and image processing. These agencies continue to sponsor image understanding research, and the various military services through their research sponsoring offices as well as the National Science Foundation have also become heavily involved in pattern recognition and image understanding research. Applications-oriented research at the US Army White Sands Missile Range (WSMR) and at the US Army Night Vision Laboratories (NVL) has lead recently to systems of reasonably high sophistication using concepts developed in-house and through sponsored research to solve complex identification and tracking problems. WSMR has concentrated on objects in the visible spectrum and in real-time, while NVL has been primarily concerned with the infrared and in near real-time. Many other systems, not necessarily real-time, have been developed for applications in medicine, meteorology, and space research.

The development of an intelligent real-time video (RTV) tracking system has been accomplished through the cooperative efforts of research and development personnel at WSMR, New Mexico State University (NMSU), and the Optical Sciences Center (OSC) of the University of Arizona. The prototype RTV processor is being assembled at NMSU, the automatic zoom lens and image rotator at the University of Arizona, and the system interfaces at WSMR. The system components will be integrated and the system deployed early in fiscal year 1979 as an add-on modification to the Contraves Model F cinetheodolite at WSMR.

This report contains several sections which describe the RTV tracking system and present the results of research related to the development and evaluation of an intelligent video tracker. These sections are listed below:

SECTION 1. A REAL-TIME VIDEO TRACKING SYSTEM

This section presents a concise description of the RTV tracking system with its distributive array of high-speed processors. Much of the information for this section was obtained from the annual report[1] submitted by NMSU for contract DAAD07-77-C-0046.

[1] Flachs, G. M., P. I. Perez, R. B. Rogers, S. J. Szymanski, J. M. Taylor, and Yee Hsun U, "A Real-Time Video Tracking System," Annual Report for Contract DAAD07-77-C-0046, WSMR, January 1978.

SECTION 2. COMPUTER SIMULATION OF THE REAL-TIME VIDEO TRACKER

This section describes the simulation program and its utility in both design and evaluation of real-time processing and tracking algorithms. Representative results are included.

SECTION 3. TOOLS DEVELOPED AT WSMR FOR OPTICAL TRACKING RESEARCH

The hardware and software which comprise the WSMR image processing laboratory are described in this section.

SECTION 4. RESULTS OF WSMR IMAGE PROCESSING RESEARCH

The following appendices are also attached to the report:

APPENDIX I. "PATTERN RECOGNITION AND REAL-TIME BORESIGHT CORRECTION--A TUTORIAL"

This appendix is an invited paper written by Alton L. Gilbert and presented at the SPIE Seminar-in-Depth on Photo- and Electro-Optics in Range Instrumentation, 13-14 March 1978, Fort Walton Beach, Florida.

APPENDIX II. NOVEL CONCEPTS IN REAL-TIME OPTICAL TRACKING

This appendix is a paper written by Alton L. Gilbert and Michael K. Giles for presentation at the Army Science Conference, 20-22 June 1978, US Military Academy, Westpoint, New York.

SECTION 1. A REAL-TIME VIDEO TRACKING SYSTEM

An intelligent RTV tracking system will be deployed early in fiscal year 1979 as an add-on modification to the Contraves Model F cinetheodolite at WSMR. The intelligence of the RTV tracker is contained in the RTV processor. Figure 1.1 is a block diagram of the RTV tracking system which shows the RTV processor as the central element. The RTV processor receives standard composite video from a television camera, locates the target image, and provides control signals which drive the zoom and image rotator elements and point the Contraves tracking optics at the target. It also provides boresight correction signals and target attitude angles which are recorded into the vertical retrace period of the video tape used to record the tracking sequence.

A RESEARCH-ORIENTED PROCESSOR CONFIGURATION

Each of the four high speed distributive microprogrammable processors which comprise the RTV processor requires a stored microprogram to control its designated tracking function. To provide a powerful tool for future research in video tracking algorithms and to facilitate operational testing of the RTV system, the control store of each processor is realized with a read/write random access memory. A minicomputer-based input/output processor is used as a programmable interface among the user, the four distributive processors, and the video tape recorder (VTR) system. This flexible, reprogrammable structure is illustrated by the input/output processor configuration of figure 1.2.

The input/output processor architecture consists of a PDP 11/35 minicomputer with associated peripherals and interfaces which allow the user to load, store, edit, and debug the control programs for each of the other four distributive processors; to monitor and display the real-time performance of the RTV system; and to record the tracking data on video tape during each vertical retrace period. The Tektronix 4014 graphics display terminal provides an excellent user terminal for controlling and evaluating the performance of the RTV tracking system. The telephone modem and floppy disk provide the capability to load and store the tracking programs from the computer system at NMSU.

A STANDARD MICROPROGRAMMABLE PROCESSOR ARCHITECTURE

The four distributive processors are being built with a standard microprogrammable processor architecture to simplify the development and maintenance of the RTV tracking system. This standard architecture, shown in figure 1.3, has been designed, built, and tested at NMSU.

Based on the new Texas Instruments (TI) 74S481 Schottky processor chip, it provides a microinstruction cycle time of under 200 nanoseconds with sufficient computational power to implement the required RTV tracking algorithms. The standard architecture requires several LSI chips which may be partitioned into control and processing sections.

The control section consists of a Signetics 8X02 microinstruction address sequencing chip which provides a microprogram counter (MPC), a 1KX48 read/write random access memory which serves as a control store (CS) for the microinstructions used to implement the tracking algorithms, and an edge triggered D flip/flop microinstruction register (MIR) which holds the currently executing microinstruction while the next one is being accessed in the CS. Overlapping the execution of one microinstruction with the fetch of the next one allows the processor to achieve a minimum microinstruction cycle time equal to the larger of either the fetch time or the execution time, significantly increasing the speed of the processor.

The processing section consists of a cascadable array of TI 74S481 4-bit slice processing elements, a register file comprising an array of high-speed registers used by the processor under program control for temporary storage of addresses and data, input and output flag circuitry, and an external bus system. The input and output flags constitute a set of processor status flags which may be tested under microprogram control. In addition, some flags are used for communication between processors or for testing signals external to the system.

The four processors which comprise the RTV processor are described in some detail in the following paragraphs. In each case, the processor is built around the standard architecture described above. Some specialized hardware is added to the standard configuration in each case to accommodate the specific functions of the individual processors.

THE VIDEO PROCESSOR

The video processor decomposes each video field into target, plume, and background pixels at the standard video rate of 60 fields per second. As the TV camera scans the scene, the video intensity is digitized at m equally spaced points across each horizontal scan line. A resolution of m = 512 pixels per line results in a pixel rate of 96 nanoseconds per pixel. Within 96 nanoseconds, a pixel intensity is digitized and quantized into eight bits (256 gray levels), counted into one of six 256-level histogram memories, and then converted by the decision memory to a 2-bit code indicating its classification (target, plume, or background). The 2-bit classification code is passed to the projection processor via the target data (TD) and projection data (PD) lines. TD is high for target points; PD is high for plume points. This sequence

of pixel operations is illustrated by the upper path of the block diagram of figure 1.4.

The sync stripper (figure 1.4) provides a video-derived sync signal which synchronizes the pixel clock at the beginning of each video line and provides vertical timing pulses for the region definition logic and the other distributive processors. The pixel clock provides the sample clock for the A/D converter and the horizontal timing pulses for the region definition logic. The region definition logic converts the horizontal timing pulses to target and plume strobes (TS and PS) which are transmitted to the projection processor with an appropriate delay for pixels located within the target and plume windows described below.

During the vertical retrace period (approximately 1.2 msec) the video processor applies the decision algorithm contained in its control store to the stored histograms of the field just digitized and then sets the appropriate bits in each of the 256 words of decision memory to allow real-time classification of the next field of pixels. Each 2-bit word of the decision memory represents one of the 256 possible levels of pixel intensity. During the real-time pixel classification operation, the two bits of a given word are connected to the TD and PD lines whenever a pixel intensity corresponds to the address of that word in the decision memory.

Toward the end of the vertical retrace period, the video processor receives the computed location of the top left corners, the heights and widths of the target and plume tracking windows from the tracking processor via the communications memory (figure 1.4), and loads this data into the region definition logic for use during the next field. A set of counters in the region definition logic are preset with this information just before digitization begins and decremented by the sync and pixel clock signals to provide timing for the vertical and horizontal extent of the tracking windows. Although the entire field-of-view (FOV) of the TV camera is digitized at 60 fields per second, only those pixels which lie within the target and plume windows are classified, counted, and processed by the rest of the system.

The basic assumption of the image decomposition method is that the target image has some video intensities not contained in the immediate background. A tracking window is placed about the target image, as shown in figure 1.5, to sample the background intensities immediately adjacent to the target image. The window frame is partitioned into two regions, B and P. Region B is used to provide a sample of the background intensities, and region P is used to sample the plume intensities when a plume is present. Using the sampled intensities, a very simple decision rule is used to classify the pixels in region T as follows:

- Background points--All pixels in region T with intensities found in region B are classified as background points.

- Plume points--All pixels in region T with intensities found in region P, but not found in region B, are classified as plume points.

- Target points--All pixels in region T with intensities not found in either region B or P are classified as target points.

The six histogram memories used in the video processor accumulate intensity histograms of the three regions (B, P, and T in figure 1.5) within each of two independent tracking windows. The control store may be programmed to implement more complex decision rules, when appropriate, by fully exploiting the statistics of these intensity histograms. For example, learned estimates of the probability density functions for target, plume, and background intensities can be obtained from the measured histograms, and a Bayesian classifier can be used to decide whether a given pixel should be classified as a target, plume, or background point.

A tracking window placed about the target image provides a method for sampling the pixel features associated with the target and background images. The background sample should be taken relatively close to the target image, and it must be of sufficient size to accurately characterize the background intensity distribution in the vicinity of the target. The tracking window also serves as a bandpass filter by restricting the target search region to the immediate vicinity of the target. Although one tracking window is satisfactory for tracking missile targets with plumes, two windows provide additional reliability and flexibility for independently tracking a target and plume, or two targets. Figure 1.6 shows typical tracking situations with two tracking windows. Having two independent windows allows each to be optimally configured and provides reliable tracking when either window can track.

If the target to be tracked requires only one window, then the other window can be expanded to include the entire FOV as shown in figure 1.7. The outer window provides additional reliability since it can locate the target image as long as it is in the optics FOV. However, the outer window is subject to more noise due to its larger size.

THE PROJECTION PROCESSOR

The projection processor consists of a projection accumulation memory (PAM) and a standard processor which are designed to form projections of simultaneous target and plume windows and to compute structural parameters from the projections. The pixel data from each tracking

window enters the PAM in real-time as a synchronized serial stream on lines TD and PD. As the classified pixel data is received, the PAM accumulates the projection data while the processor monitors the y-projections, accumulates the total number of target and plume points, and determines the midpoints used to split the x-projections. Each x-projection is split to allow the computation of target and plume attitude angles based on the locations of the median centers of the x- and y-projections of the top half and bottom half of the target and plume images. In the vertical retrace interval, the processor assumes addressing control of the PAM and computes the structural parameters from the projection data. A block diagram of the projection processor is shown in figure 1.8.

Each incoming pixel is assigned a memory location in the appropriate 1536 x 9 random access memory (RAM). A separate memory location is preassigned for each row and column of each tracking window. Target and plume windows containing up to 511 x 511 pixels can be accommodated. The first row of pixels in a given window is used to clear the memory locations assigned to the vertical columns of pixels to initialize the accumulation of the y-projection for that row. Once the memory locations for the columns are cleared, the value of the next pixel which occurs in each column is added to zero and the result is written back into the same memory location for that column by an arithmetic logic unit (ALU). The value (1 or 0) of each subsequent pixel; i.e., the value of TD or PD for each pixel, is added to the contents of the appropriate memory location by the ALU used to accumulate the x-projections. Similarly, after the memory location corresponding to a given horizontal row is cleared by the first pixel occurring in that row, the value of each subsequent pixel occurring in the row is added to the contents of the memory location to accumulate the value of the y-projection for the row. A separate ALU and high-speed RAM are used for each projection to accumulate both horizontal and vertical projections simultaneously at pixel rates of 11 MHz or greater.

The processing portion of the projection processor uses the standard processor with five 4-bit slices and an expanded register file to accommodate targets or plumes consisting of more than 32k points. As the PAM accumulates the projections for a given line of the target window, the total number of target points in that line (the y-projection) is multiplied by two and stored by the processor. The same is true for lines in the plume window. All lines following the first active line of the window are processed in the same manner, and the total number of target points times two is accumulated. If at the end of a line this number is greater than the total number of target points in the previous frame, the top x-projection is terminated. A flag is sent to the PAM forcing the x-projection for the next line to be placed at the starting address of the bottom x-projection. Thus, three separate projections

are accumulated for each window--a y-projection, and top and bottom x-projections. These six projections, stored in the two 1536 x 9 RAMs at the end of each field, are mapped into a continuous address space at the beginning of the vertical retrace period to allow easy access by the projection processor.

During the vertical retrace interval, the projection processor divides each projection into eight segments of equal mass using a simple algorithm to sequentially address each line of the projection and multiply the number of pixels in the line by eight. If the result exceeds the total number of pixels in the projection, a flag is sent to the PAM forcing the next line to be placed at the beginning of the next 1/8 segment of the projection. If the result is less than the total number of pixels in the projection, additional lines of pixels are accumulated until the line containing the 1/8 percentile point is located.

The A and B ports of the processor (see figure 1.8) are used directly to perform the multiplications required to respectively split the x-projections and find the 1/8 percentile points for each projection. Data read in on the A-bus is multiplied by two with a hard-wired shift, while data read in on the B-bus is multiplied by eight in the same manner.

The 1/8 percentile points for each of the six projections shown in figure 1.9 are computed within 410 μsec of the vertical retrace period and then passed to the communication memory along with the total number of target and plume points. These parameters constitute the structural parameters used by the tracker processor to define an intelligent tracking strategy.

TRACKER PROCESSOR

The tracker processor receives the structural parameters from the projection processor, locates and characterizes the structure of the target and plume images, and decides on a tracking strategy to maintain track. It then outputs control signals to place the window frames in the video processor and outputs target location and orientation data to the control processor along with a confidence in the measured data. Since it operates on the projection data from field n while the projections for the next field (n+1) are being accumulated, the tracker processor is always one field behind the video and projection processors. The tracker and control processors must both finish their calculations before the vertical retrace interval begins for field n+1. This constraint requires the tracker processor to output its data to the control processor within 7 milliseconds after it receives the projection data.

Conceptually, the tracking algorithm can be viewed as a finite state Mealy machine defined by the following next state and output equations:

$$q_n = \delta\,(q_{n-1},\ i_{n-1},\ i_{n-2},\ \ldots,\ i_{n-k}) \tag{1.1}$$

$$z_n = \omega'(q_n,\ i_n,\ i_{n-1},\ \ldots,\ i_{n-k}) \tag{1.2}$$

$$z_n = \omega\,(q_{n+1},\ i_n,\ i_{n-1},\ \ldots,\ i_{n-k}) \tag{1.3}$$

Here q_{n+1} is an interpretation of the preset FOV situation by the tracking algorithm based upon the previous interpretation and the sequence of inputs $i_n, \ldots, i_{n-k}$ corresponding to the input data from the projection processor for frames n, ..., n-k, respectively. The output z_n is the tracking strategy given to the video and control processors in response to the inputs. δ and ω represent the next state and output mappings, respectively.

A target with a plume is treated as two distinct objects by the projection processor which outputs two disjoint sets of data. The tracking algorithm processes the next state mapping for each object and determines the best response strategy to maintain track. This is illustrated by the following machine equations:

$$i_n = i_n^T \cup i_n^P \tag{1.4}$$

$$q_n^T = \delta^T(q_{n-1}^T,\ i_{n-1}^T,\ \ldots,\ i_{n-k}^T) \tag{1.5}$$

$$q_n^P = \delta^P(q_{n-1}^P,\ i_{n-1}^P,\ \ldots,\ i_{n-k}^P) \tag{1.6}$$

$$z_n = \omega(q_{n+1}^T,\ q_{n+1}^P,\ i_n,\ \ldots,\ i_{n-k}) \tag{1.7}$$

Since the tracker processor is the only processor that communicates with all of the other three processors, each of which has its own coordinate system, the tracker processor must interpret the input data intelligently and then output the appropriate data to the video and control processors in their respective coordinate systems. The 44 inputs presented in figure 1.10 are positive 16-bit integers defined for a coordinate system whose origin is the first pixel scanned inside the appropriate tracking window. The eight outputs to the video processor presented in figure 1.11 are 9-bit positive integers defined for a coordinate system whose origin is the first pixel scanned within the FOV. Of the seven 16-bit outputs to the control processor listed in figure 1.12, DX, DY, DRX, and DRY are integers defined for a coordinate system whose origin is the boresight. The left-most bit (bit 15) is used as the sign bit for each word.

STATE is also in the integer format. A binary point is assumed between bits 15 and 14 for WGHT, and between bits 11 and 10 for DZ.

The angle from vertical boresight included in figure 1.12 is computed directly from the projection data using the projection median technique illustrated by figure 1.13. The projection median technique dissects a binary image into two segments. The x- and y-projections are accumulated for each segment and the median coordinates TXT, TXB, TYT, and TYB defined in figure 1.10 are evaluated for each projection. The rotational angle θ_{PM} with respect to the y-axis is then defined as the angle between the y-axis and the line joining the two median centers.

$$\theta_{PM} = \arctan \frac{TXT - TXB}{TYT - TYB} \tag{1.8}$$

There are two main advantages for using the median center instead of the geometric centroid to determine the angle of rotation. First, the medians are easier to compute, and the projections used to compute the medians provide additional information; such as, length-to-width ratio and target shape. Secondly, the median is less sensitive to noise perturbations because the distance of the noise pixels from the target is not used in the median computation.

With the orientation known, other image position information such as the target tail or the plume tip can be easily computed from the projections. When the target is being tracked, the target tail is used to compute the x- and y-displacements from boresight. To allow a smooth transition from target window to plume window tracking; i.e., when the target becomes too small to track, the plume tip is used to compute the boresight displacements.

An overall view of the functions of the tracker processor is given in figure 1.14. It has two modes of operation, the initial acquisition mode and the autotrack mode.

The initial acquisition mode is used when the RTV system is trying to lock onto the target of interest. During this mode, the video processor does little or no learning on the target and plume intensities. The tracker processor will not instruct the control processor to begin predicting the target location until it is sure of the existence of at least the plume within the plume window. When the plume image moves into an appropriate region of the FOV, the tracker processor will notify both the video processor and the control processor with a flag indicating that it is now ready to shift into the autotrack mode.

The autotrack algorithm is divided into the four main modules shown in figure 1.14. The data conversion module transforms the projection input

data into physical variables; such as, target and plume size, position, and shape. These variables are then combined with previous target activity data from the history update module to obtain additional variables; such as, the changes in target and plume position and size. All of these variables are compared with preassigned reference constants to obtain a set of binary inputs which are used directly by the state interpretation module (see table 1.1) to define the current tracking situation and produce an optimum tracking strategy. The strategy is implemented by the output computation module in the form of the control signals to the video and control processors described in figures 1.11 and 1.12.

The tracking situations of both the target and plume images are described by four states which, when combined, form a total of 16 system tracking states. Table 1.2 lists the four image states and eight examples of the system tracking states.

The basic hardware architecture of the tracker processor consists of the standard TI 74S481 microprogrammable processor and a 2k x 16 RAM. This working memory contains the various lookup tables, data storage area, and the scratch pad area.

It is highly desirable to keep the control store of the tracking processor within 1k because of the 10 bits address bus of the 8X02 control sequencer. This constraint heavily influenced the basic design and implementation of the tracking processor algorithm, whose computation, though rather simple and straightforward, is lengthy. Lookup tables are used whenever possible to simplify the computations and reduce the microcode. At a cycle time of less than 200 nanoseconds, the standard processor is capable of executing over 35k instructions within the 7-millisecond time limit which is more than enough for the current algorithm.

The state interpretation module is implemented as a lookup table because of its finite state machine approach. The target inputs are encoded into 10 bits as in the example of table 1.1, and the plume inputs are encoded similarly into 7 bits. By using the inputs as an address, the state interpretation module is implemented as a 1k by 16 lookup table in the working memory. A state transition is effected when the set of target inputs for a given pixel field addresses the proper memory location in the lookup table. The optimum next state for that set of target inputs is read out of the field of 2 bits selected by the current target state as shown in figure 1.15. Similarly, the set of plume inputs and the current plume state are used to select the optimum next plume state, and the new plume and target states are combined to define the optimum tracking state to be implemented during the next video field.

The zoom correction algorithm is also computed by means of a lookup table. The use of lookup tables for implementing algorithms has many advantages.

TABLE 1.1. EXAMPLE BINARY TARGET INPUT DATA TO THE STATE INTERPRETATION MODULE

Bits	Binary Values	Target Condition
0	1	Target too long
1 and 2	10	Too many target points
3 and 4	10	Target size increased too much
5	1	Correct target shape
6	1	Random target motion
7	1	Target within field-of-view
8	1	Trackable target location
9	1	Drastic change of target location

TABLE 1.2. THE FOUR IMAGE STATES AND EIGHT OF THE 16 SYSTEM TRACKING STATES

Image States	Description
00	Normal tracking
01	Abrupt change
10	Out of FOV
11	Lost

State	Target		Plume		Description
S1	0	0	0	0	Normal tracking of both images
S2	0	0	0	1	Target tracking, abrupt plume change
S3	0	0	1	0	Target tracking, plume out of FOV
S4	0	0	1	1	Target tracking, plume lost
S5	1	1	0	0	Target lost, plume tracking
S6	1	1	0	1	Target lost, abrupt plume change
S7	1	1	1	0	Target lost, plume out of FOV
S8	1	1	1	1	Target lost, plume lost (use trajectory data)

It significantly reduces the amount of required control store, but more importantly, it allows the next state mapping; hence, the response strategy, to be easily modified to suit other forms of images without changing the basic algorithm in the microcodes.

THE CONTROL PROCESSOR

The function of the control processor is to generate the four control signals that drive the real-time video tracker; i.e., the tracker azimuth A_i and elevation E_i which are sent to the RTV-Contraves system interface and the optics rotation ϕ_i and zoom Z_i which are sent to the RTV-zoom/rotation interface (figure 1.1). In addition, the control processor outputs the following tracking data to the input/output processor after each field so that they can be recorded in the vertical retrace period of the video tape: field count, tracker status, time, x-displacement from boresight, y-displacement from boresight, tangent of the target orientation angle from vertical boresight, target azimuth, target elevation, tracker azimuth, tracker elevation, image rotation angle, and zoom ratio.

The tracking optics feeds the target image to the video processor portion of the RTV processor (figure 1.1) which establishes the target coordinates with respect to the optics boresight. The control processor combines current target coordinates with previous target coordinates to point the optics toward the next expected target position. Radar derived tracking data is also available to the control processor from the WSMR precision acquisition system (PAS).

There are several inherent differences between the optical and the PAS data. First, the optical system tracks the target in real-time with 60 updates per second using a 2-dimensional pointing vector, while the PAS tracks the target in 3-dimensional space with a 200 millisecond delay and 20 updates per second. Secondly, when the target is visible, the optics data should be an order of magnitude more precise than the PAS data. The PAS data, however, is available when the optical data is lost due to clouds or poor visibility. The control processor must take these differences into consideration and generate the "best" estimate control signals to point the optics towards the target. The confidence weight WGHT and the tracking STATE inputs from the tracker processor allow the control processor to properly weight the optical data in the control equations. Since the optical data is generally more precise, it is used as the primary tracking data. A manual override capability is also provided to reacquire track if necessary.

Using the target position (DX, DY), target rotation (DRX, DRY), and zoom correction (DZ) inputs from the tracker processor (figure 1.12) together with the actual tracker position (A_0, E_0), image rotation (ϕ_0), and zoom

ratio (Z_o) from encoders on the tracking mount (figure 1.1), image rotator and zoom lens respectively, the control processor computes measured values for the four control signals (A_m, E_m, ϕ_m, Z_m). The following measured control equations are implemented by the control processor:

$$Z_m = Z_o\ (1 + DZ) \tag{1.9}$$

$$\phi_m = \phi_o + \text{ATAN}\ (\frac{DRX}{DRY}) \tag{1.10}$$

$$E_m = E_o + [\frac{DY}{Y} \cos \phi_o - \frac{DX}{X} \sin \phi_o]\ \frac{FOV}{Z_o} \tag{1.11}$$

$$A_m = A_o + [\frac{DX}{X} \cos \phi_o + \frac{DY}{Y} \sin \phi_o]\ \frac{FOV}{Z_o \cos E_m} \tag{1.12}$$

X and Y are the dimensions in pixel units of the video field.

The predicted control equations are based on the combination of linear and quadratic optical estimates taken from a five-deep history stack, as well as any available PAS derived estimates. Since the input data is derived from field (K-1), and the estimates are being computed during field K, the control estimates must predict ahead two time increments to provide control signals which will place the boresight at the correct position during frame K+1. Thus the linear optical estimates have the form

$$\hat{\theta}(K+1|K-1)_L = 3\theta(K-1) - 2\theta(K-2) \tag{1.13}$$

and the quadratic optical estimates have the form

$$\hat{\theta}(K+1|K-1)_Q = \frac{1}{25}\ [81\theta(K-1)-20\theta(K-2)-21\theta(K-3)+12\theta(K-4)-27\theta(K-5)] \tag{1.14}$$

A linear combination of the linear and quadratic predictors (equation 1.15) is used to predict the target location.

$$\hat{\theta}_{OPT}(K+1) = \alpha_1\hat{\theta}(K+1|K-1)_L + \alpha_2\hat{\theta}(K+1|K-1)_Q \tag{1.15}$$

The linear coefficients α_1 and α_2 are determined on the basis of minimum variance of errors made by the predictors on their estimate of the previous video field.

When PAS data is available, $\hat{\theta}_{PAS}(K+1)$ is obtained using the 5-point quadratic predictor (equation 1.14) at the video field rates, and the linear minimum variance predictor of equation 1.16 is used to predict the target location.

$$\hat{\theta}_{COM}(K+1) = \beta_1 \hat{\theta}_{OPT}(K+1) + \beta_2 \hat{\theta}_{PAS}(K+1) \qquad (1.16)$$

The coeffieients β_1 and β_2 are determined on the basis of minimum variance of errors made by the optical and PAS predictors during the previous field.

The above estimates are computed for each of the four control signals. These signals are then passed as inputs A_i, E_i, ϕ_i, and Z_i to the control systems which respectively drive the Contraves mount, the image rotator, and the zoom lens (see figure 1.1). The control processor calculations are performed in floating-point to provide the accuracy required by the respective control systems. Hence, the fixed point and integer inputs accepted by the control processor are converted to floating-point, the calculations are performed, and then the control signals are converted back to fixed point numbers which are sent to the appropriate interfaces.

The need for a floating-point multiply and divide capability strongly influenced the decision to use the TI 74S481 as the standard processor chip. A standard processor with four 4-bit slices is used in the control processor architecture to provide a 16-bit word length which is adequate to maintain tracking control. Angle data transferred in and out of the interfaces are in integer format 16-bit circular binary. The most significant bit has a weight of 180°, the least significant bit has a weight of about 20 seconds of arc. Intermediate calculations are performed in floating-point format with 16 bits for the fraction and 16 bits for the exponent.

The control store size for the control processor microprogram is presently estimated to be about 900 48-bit control words. This does not include the lookup tables for the required function calculations. The function lookup tables will reside in RAM. These tables require an additional 512 words.

Figure 1.16 shows the random access, control store, and I/O memory required by the control processor. Also shown is the direction of data flow. In this figure, it is assumed that the function tables for the SIN(X), COS(X), and ATAN(Y,X) microroutines used in the control equations 1.9 to 1.12 will reside in random-access memory.

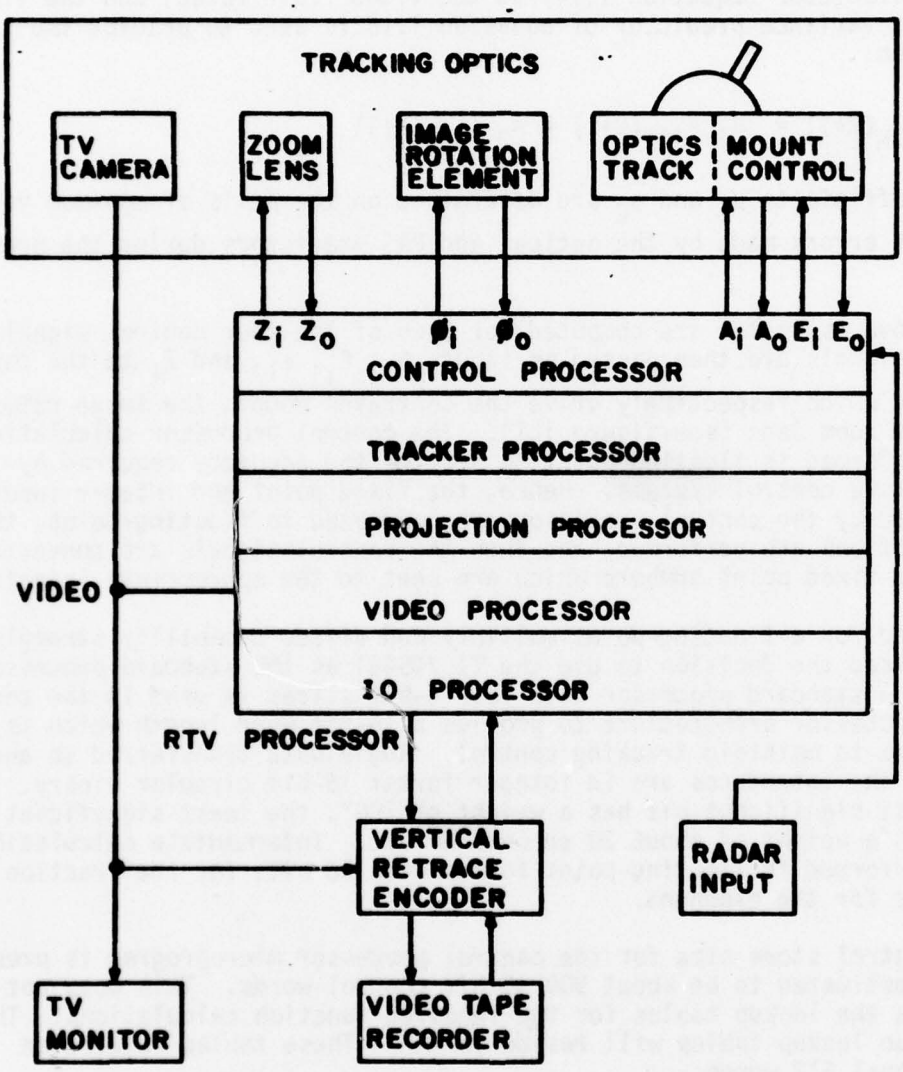


Figure 1.1. RTV Tracking System

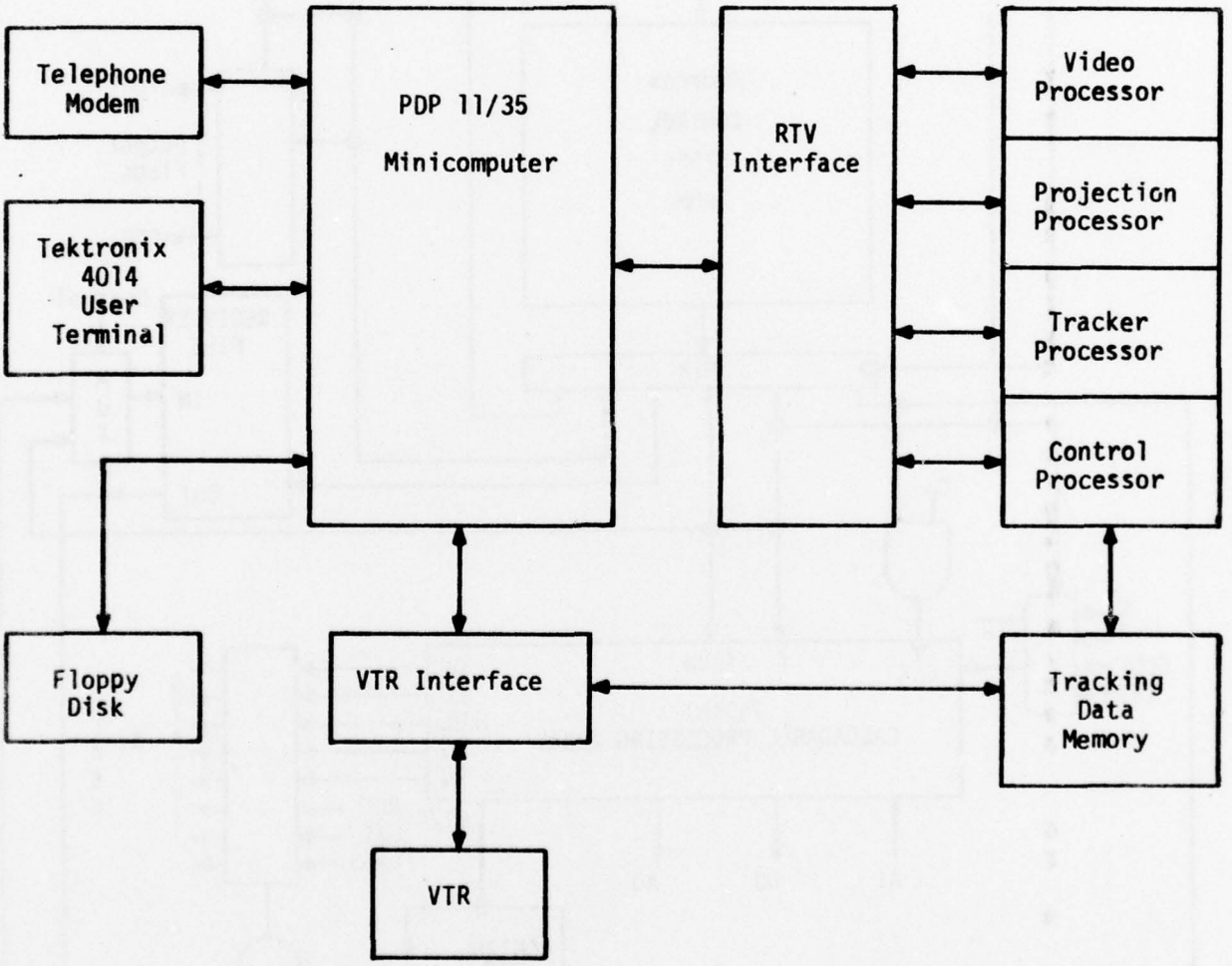


Figure 1.2. The Input/Output Processor Configuration

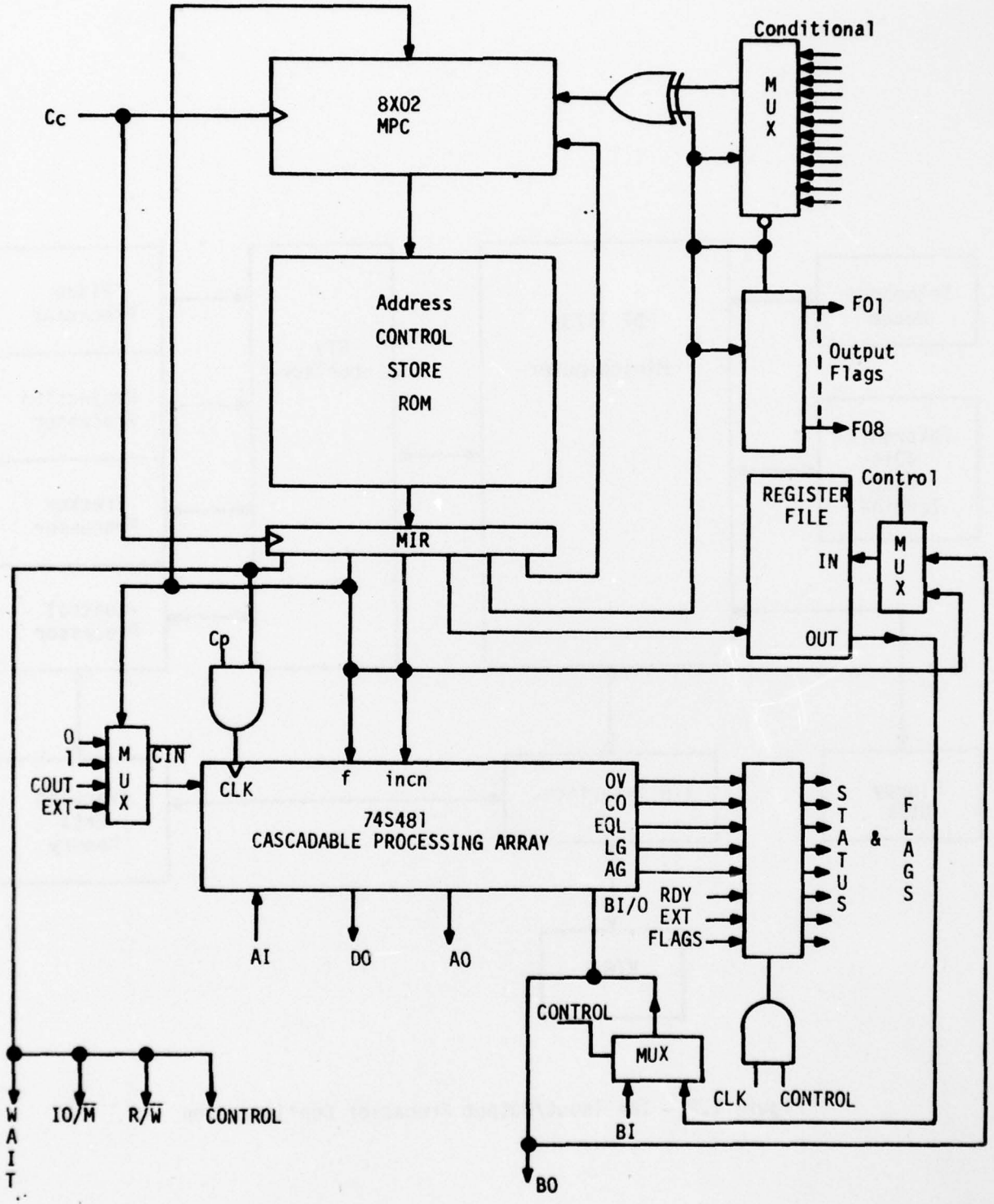


Figure 1.3. Standard Processor Architecture

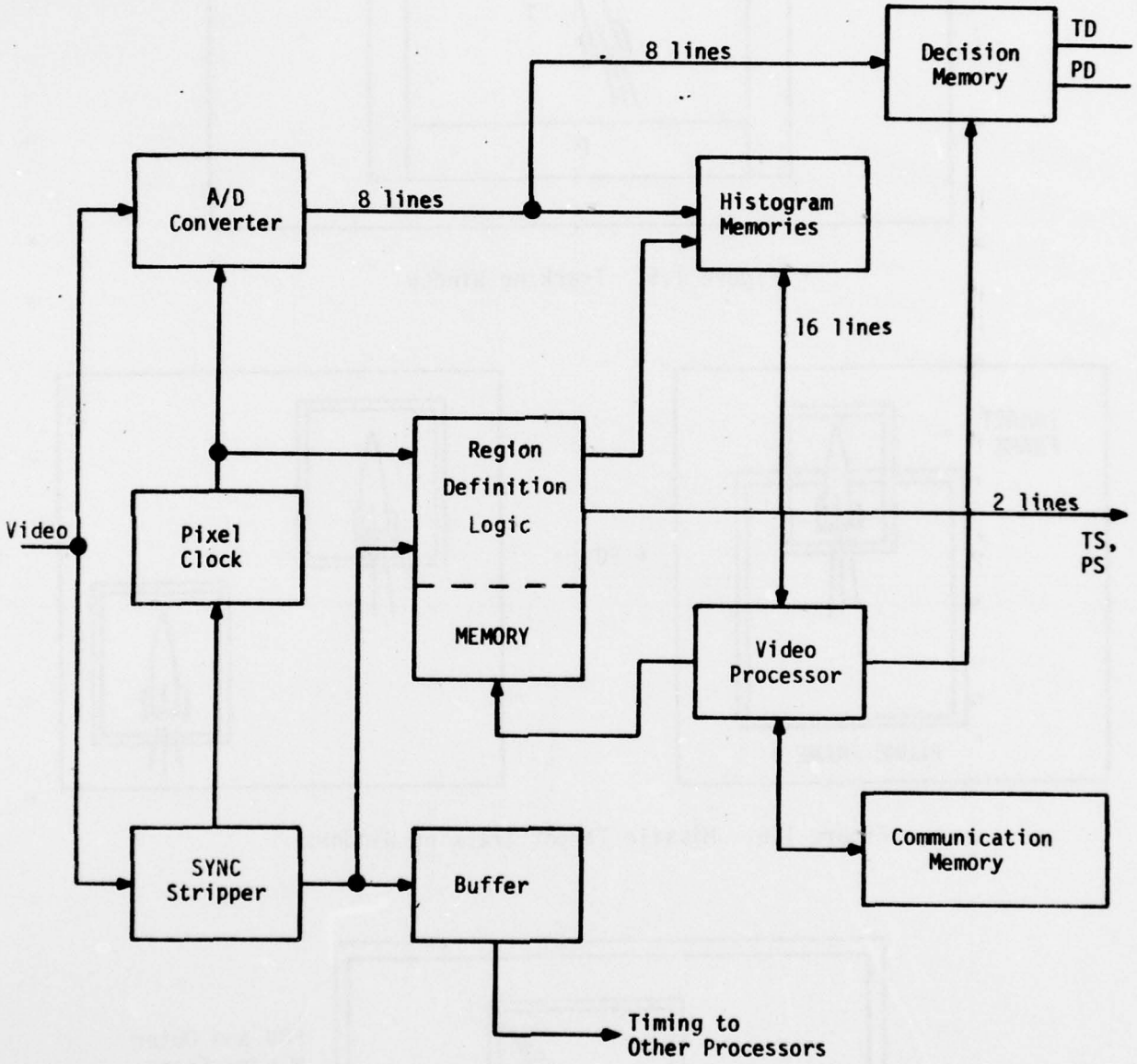


Figure 1.4. Video Processor Block Diagram

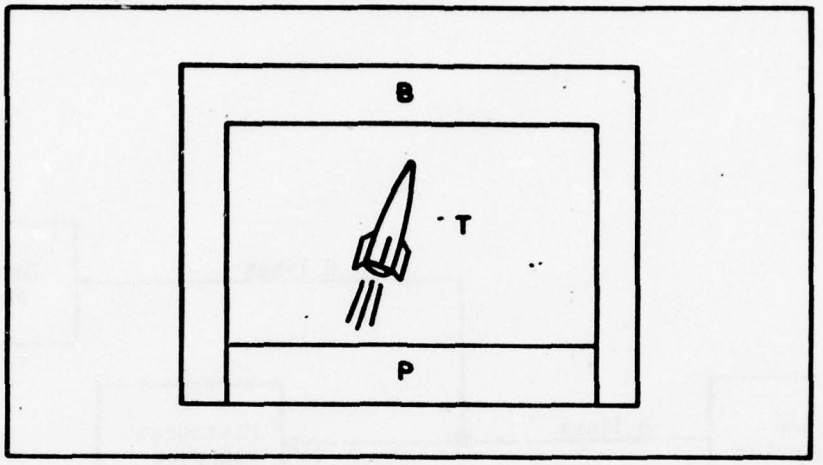


Figure 1.5. Tracking Window

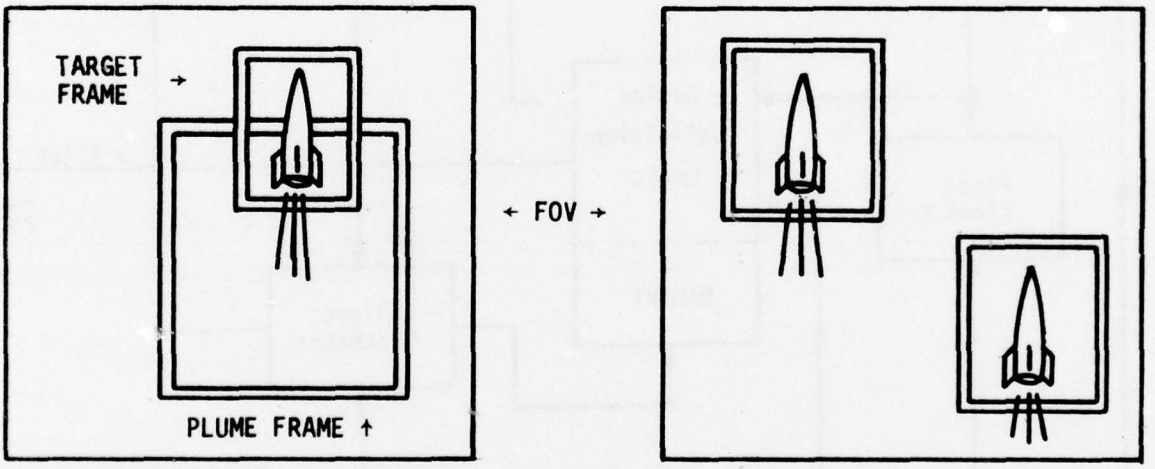


Figure 1.6. Missile Target Tracking Windows

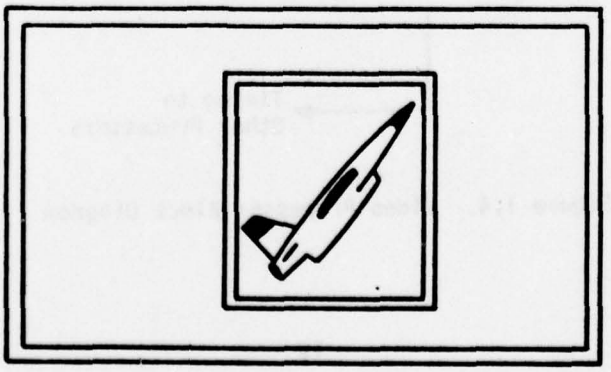


Figure 1.7. Nonmissile Target

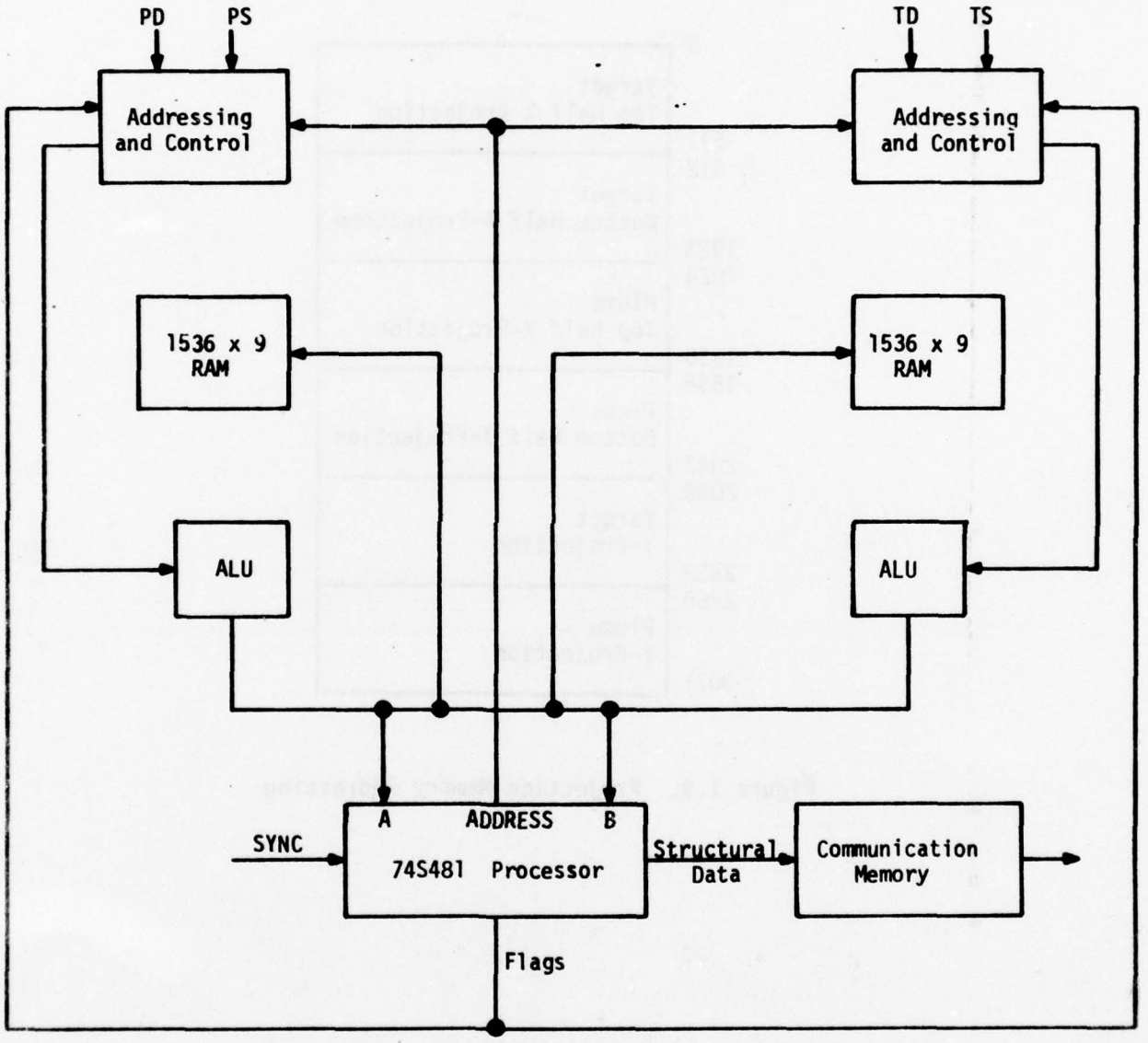


Figure 1.8. Projection Processor

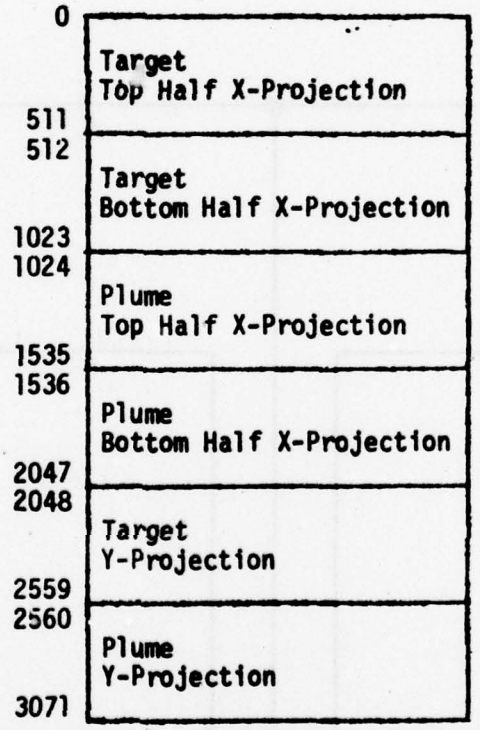


Figure 1.9. Projection Memory Addressing

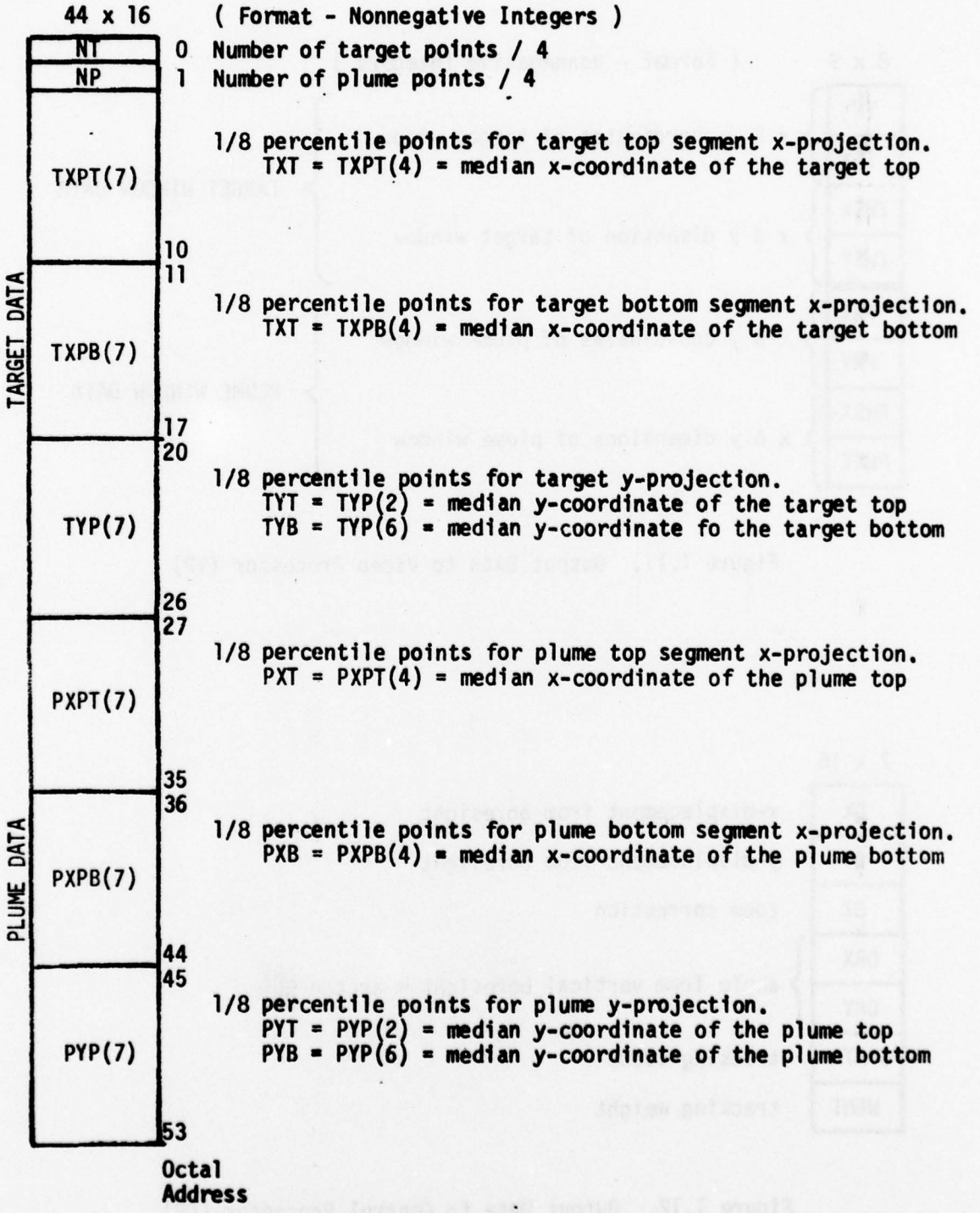


Figure 1.10. Input Data from Projection Processor

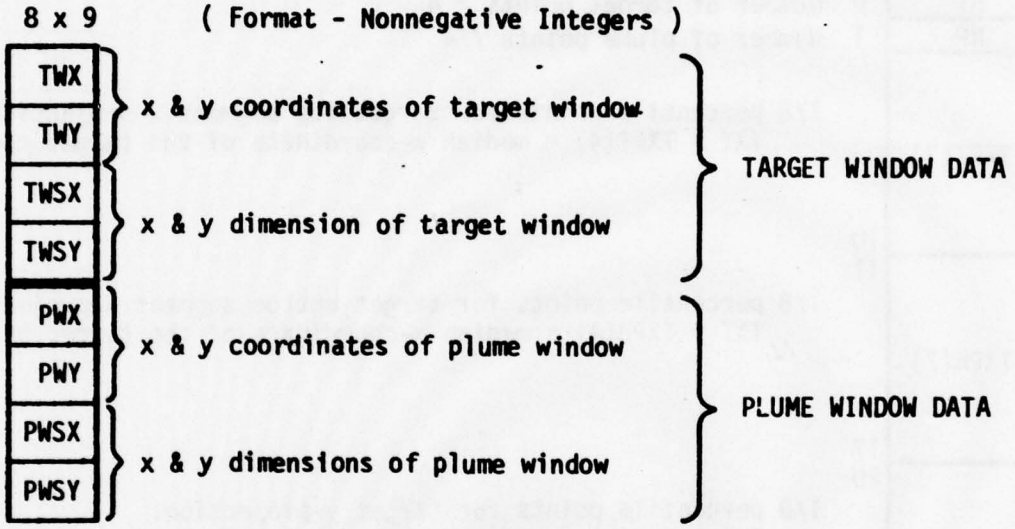


Figure 1.11. Output Data to Video Processor (VP)

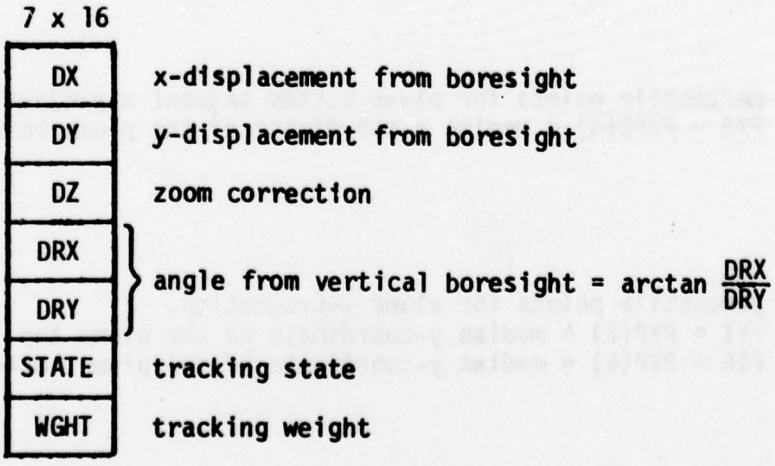


Figure 1.12. Output Data to Control Processor (CP)

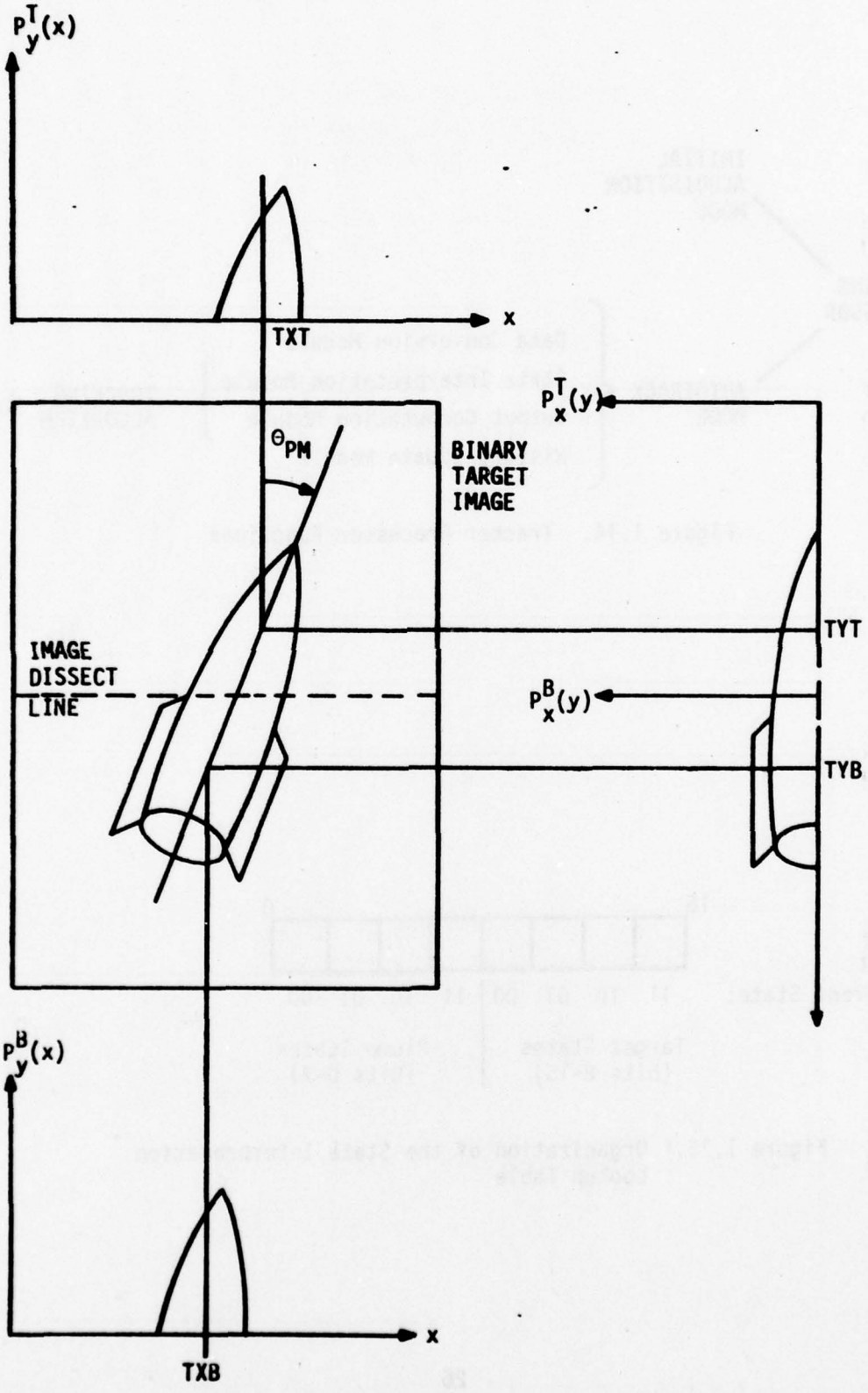


Figure 1.13. Projection Median Technique

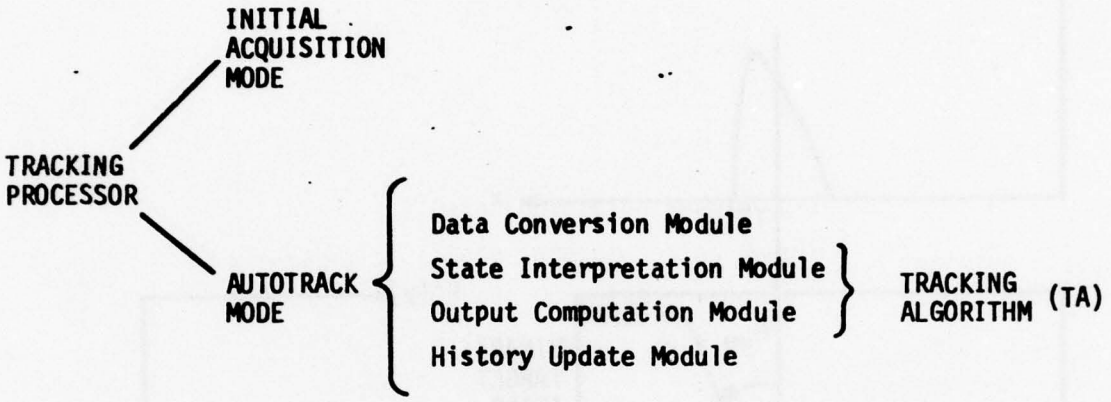


Figure 1.14. Tracker Processor Functions

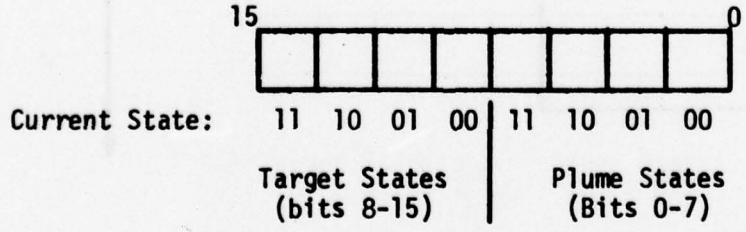


Figure 1.15. Organization of the State Interpretation Lookup Table

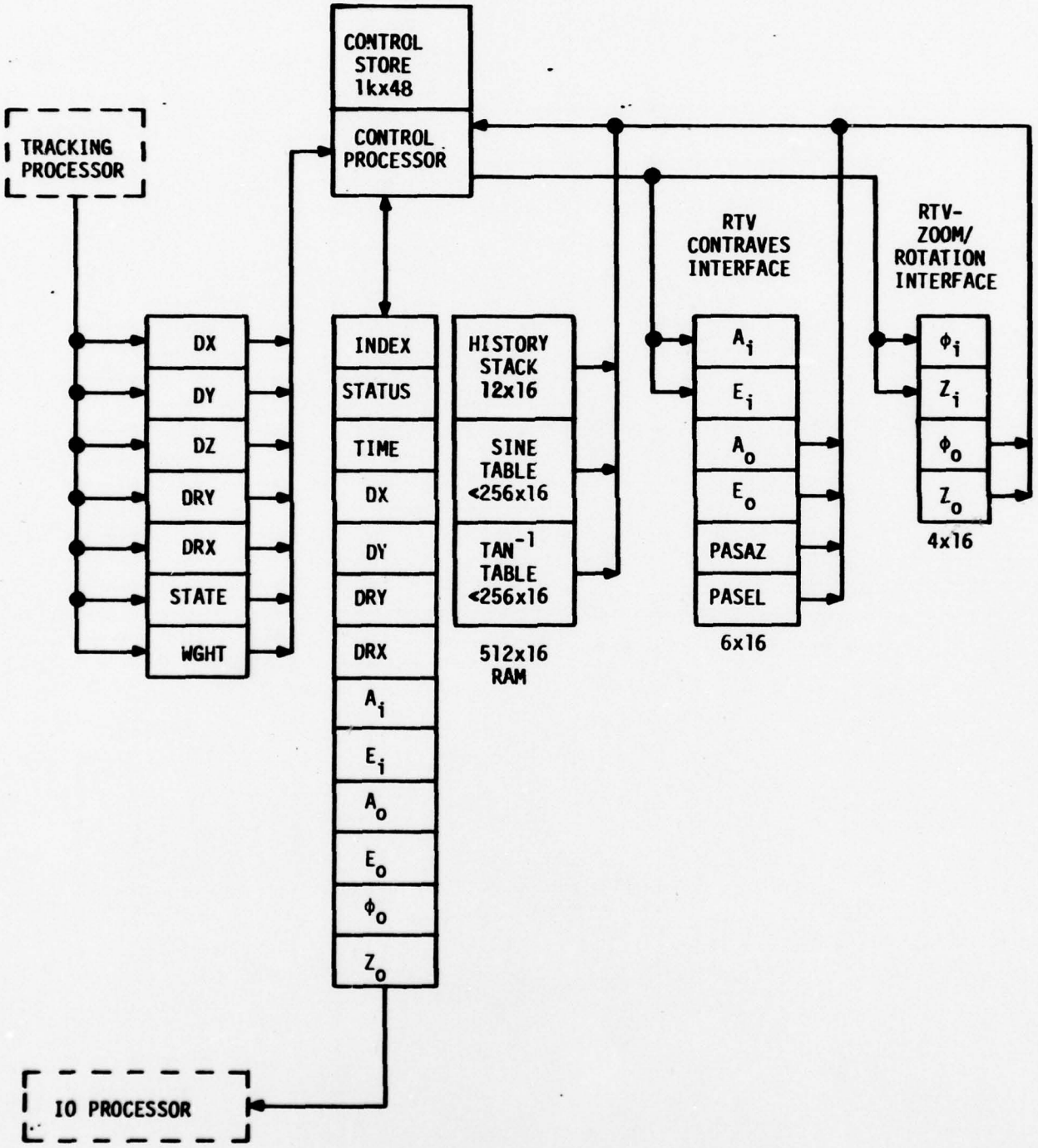


Figure 1.16. Control Processor Memory Allocation

SECTION 2. COMPUTER SIMULATION OF THE REAL-TIME VIDEO TRACKER

A computer simulation of the RTV tracking system, incorporating the algorithms used in the control stores of the four distributive processors, has been developed and implemented on the PDP 11/35 system at WSMR. A preliminary version of this program is described in detail in the 1976 NMSU final report.[2] The purpose of this simulation is to provide a method for testing new design concepts and evaluating the RTV tracking system under realistic tracking conditions. The simulation model includes dynamic models for the target trajectory and the Contraves Model F cinetheodolite tracking system, in addition to the RTV processor algorithms to simulate the complete tracking system.

The simulated target is initially located at the launch site, $\vec{L}$ = (0, 0, 0), and its trajectory is computed in three dimensions from acceleration profiles (Ax, Ay, Az) in the x, y, z orthogonal directions. The tracker is located at $\vec{T}$ = (Tx, Ty, Tz), and it is initially positioned so that the target will pass through the FOV of the tracking optics (see figure 2.1).

SIMULATION OUTPUTS

The simulation model, written in FORTRAN IV, utilizes the Tektronix 4014 graphics terminal to display the performance of the tracking system. Figure 2.2 shows the output from video field 5 of the simulation. The target and plume images are plotted as they are seen by the video processor superimposed on a crosshair which locates the boresight of the tracking optics. The control processor estimates the location of the target for the next frame and provides the control signals to place the camera boresight on the target. The performance of the control processor can be evaluated by observing the target position relative to the camera boresight.

In addition to the digitized target and plume images, the simulation displays the target and plume tracking windows and the projections. The actual and predicted target location and orientation, the current tracker boresight location and image rotation, the target view angle, and the zoom ratio are also written on the display terminal along with the time, range, and index number of the displayed field. As figure 2.2 illustrates, these outputs provide a direct measure of the performance of

2. Flachs, G. M., W. E. Thompson, R. J. Black, J. M. Taylor, W. Cannon, and Yee Hsun U, "A Pre-Prototype Real-Time Video Tracking System," Final Report for Contract DAAD07-76-C-0024, WSMR, January 1977.

each of the four processor algorithms as well as an overall measure of the tracking accuracy of the system. Optional graphics outputs designed to enhance the initial user setup of the simulation include a 3-dimensional plot of the missile trajectory and plots of the dynamic tracker azimuth, elevation, rotation, and zoom response for perfect missile trajectory data. Thus, the user can visualize how the tracker should respond and determine which tracking errors can be attributed to tracker dynamics limitations. Figure 2.3 is a 3-dimensional plot of the first 100 fields of the trajectory presently being used in the simulation. It presents a challenging tracking sequence to the RTV system with its rapidly changing direction and aspect.

In addition to its graphics display capabilities, the simulation program can dump the tracking states of up to 40 selected simulated tracking fields onto a floppy disk, thus enabling the user to restart the simulation at selected fields in the tracking sequence. Since the PDP 11/35 requires approximately 2 minutes to calculate the states for each image field, this option offers a tremendous savings in time when the user desires to observe the central or final portions of a tracking sequence.

SIMULATION INPUT DATA AND RESULTS

Two types of input are available to the simualation--simulated digitized video fields and actual digitized video fields from video tape recordings of typical WSMR tracking sequences. The simulated video data is produced by a sophisticated picture generator subroutine which generates pixel intensities belonging to target, plume, background, and foreground intensity distributions. The characteristics of these distributions were established through studies of cinetheodolite film sequences at WSMR. The picture generator not only produces simulated video, but it also projects the target and plume images onto a plane normal to the tracker boresight to simulate the proper perspective of the target and plume as seen through the tracking optics.

Figure 2.4 contains two representative simulation outputs selected from the first 100 fields of simulated digitized video. The missile trajectory is the one presented in figure 2.3. It is evident that the RTV tracker performs very well during this tracking sequence. Track is maintained throughout the sequence, and all processors appear to be functioning properly.

The recent development of an image processing laboratory at WSMR has enabled research personnel to digitize sequential video fields of typical tracking imagery. These fields of digitized video are now being used in the RTV simulation and in the development of improved image segmentation and structural analysis algorithms.

The simulation program has been modified to allow these digitized images to be input directly to the video processor in place of the simulated intensity values. This effort is just beginning. Much work remains to be accomplished both in the modification and improvement of the simulation program, and in the testing of the RTV processor algorithms with actual digitized data. The preliminary results, however, are very encouraging.

Figure 2.5(a) is a graphics display of a field of digitized video which was injected into the RTV simulation. The simulation was allowed to repeatedly process the same video field superimposed on the simulated target trajectory for several frames to test the static and dynamic responses of the RTV processors. The result after six processing frames, shown in figure 2.5(b) verifies the effectiveness of the processing algorithms used in the RTV tracker. Since no plume is present, both windows are tracking the target. Also, image rotation has not been incorporated in the simulation, but rather the tracking window is allowed to rotate and align itself with the target. Several important conclusions may be deduced from this result: The video processor successfully identifies most of the target pixels; the projection data is being accumulated and used effectively by the projection and tracker processors to obtain the target orientation and the displacement of the target from boresight; and both windows are tracking the target and closing down on it to reduce noise.

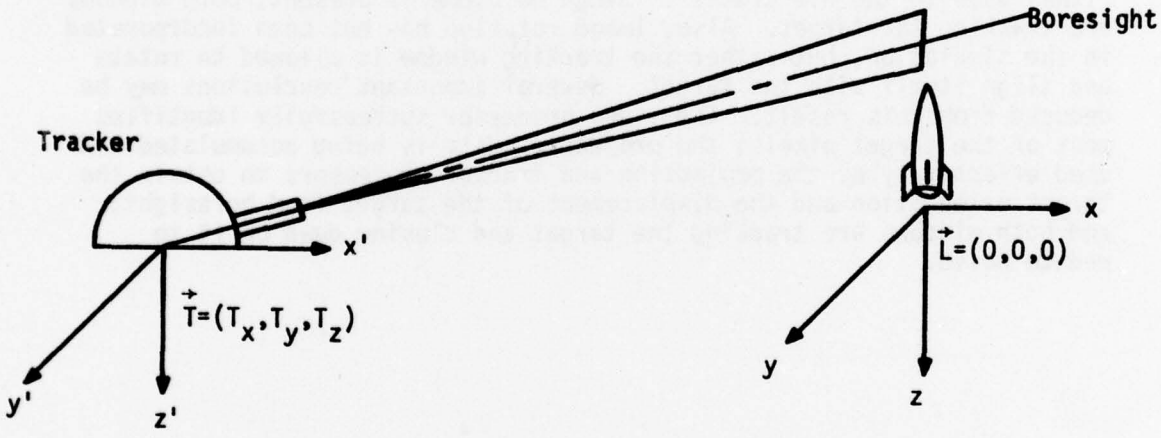


Figure 2.1. Tracking Configuration

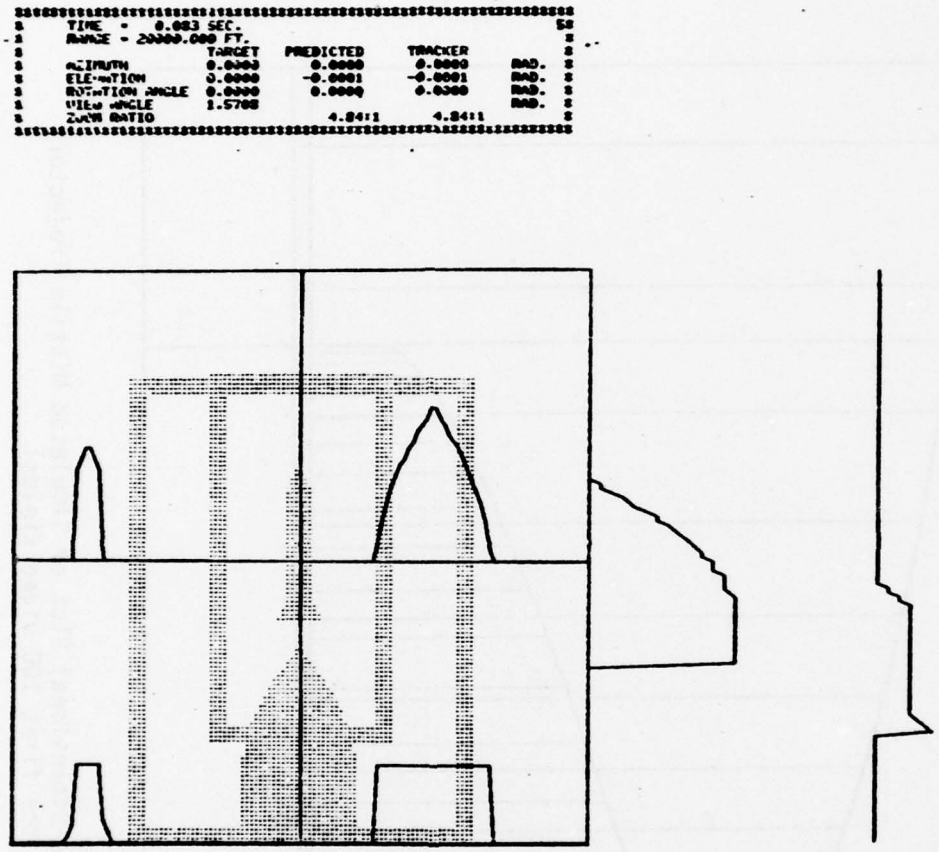


Figure 2.2. Simulation Output at Field 5

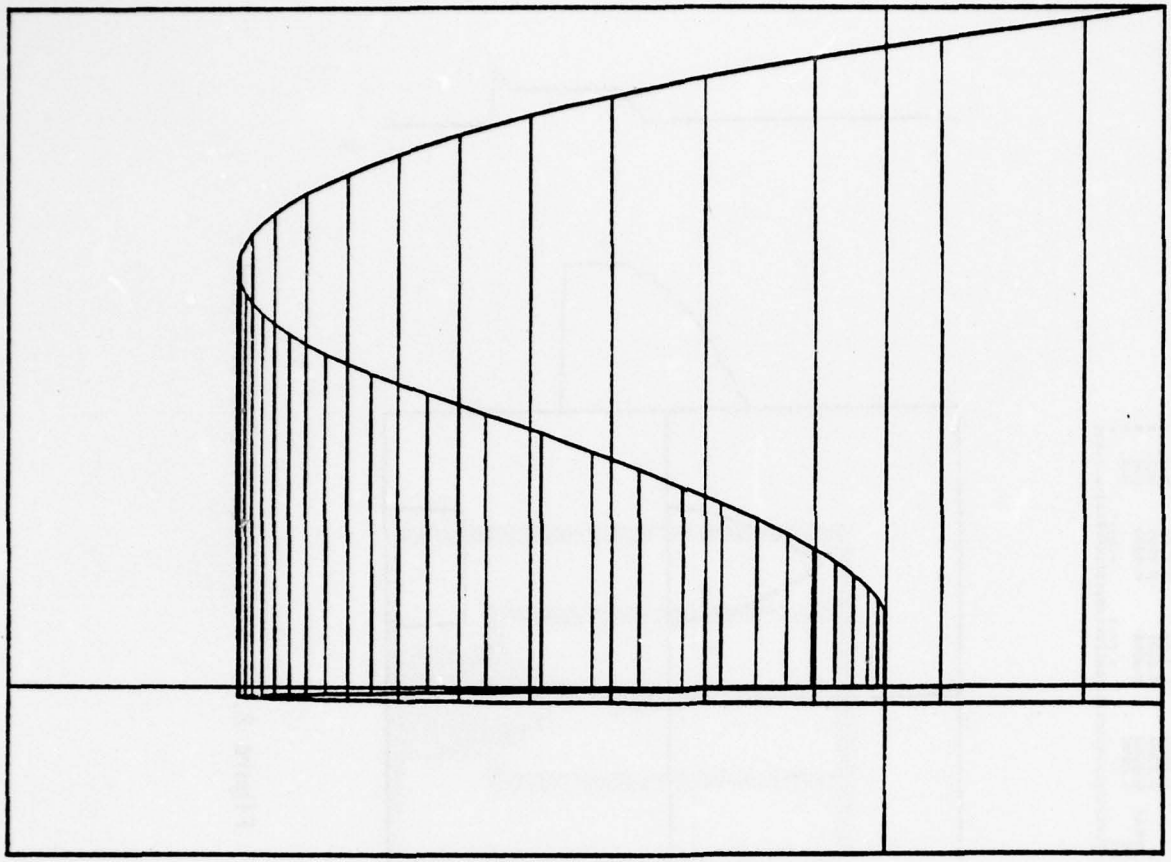


Figure 2.3. Three-Dimensional Plot of Simulated Missile Trajectory (for the first 100 video fields)

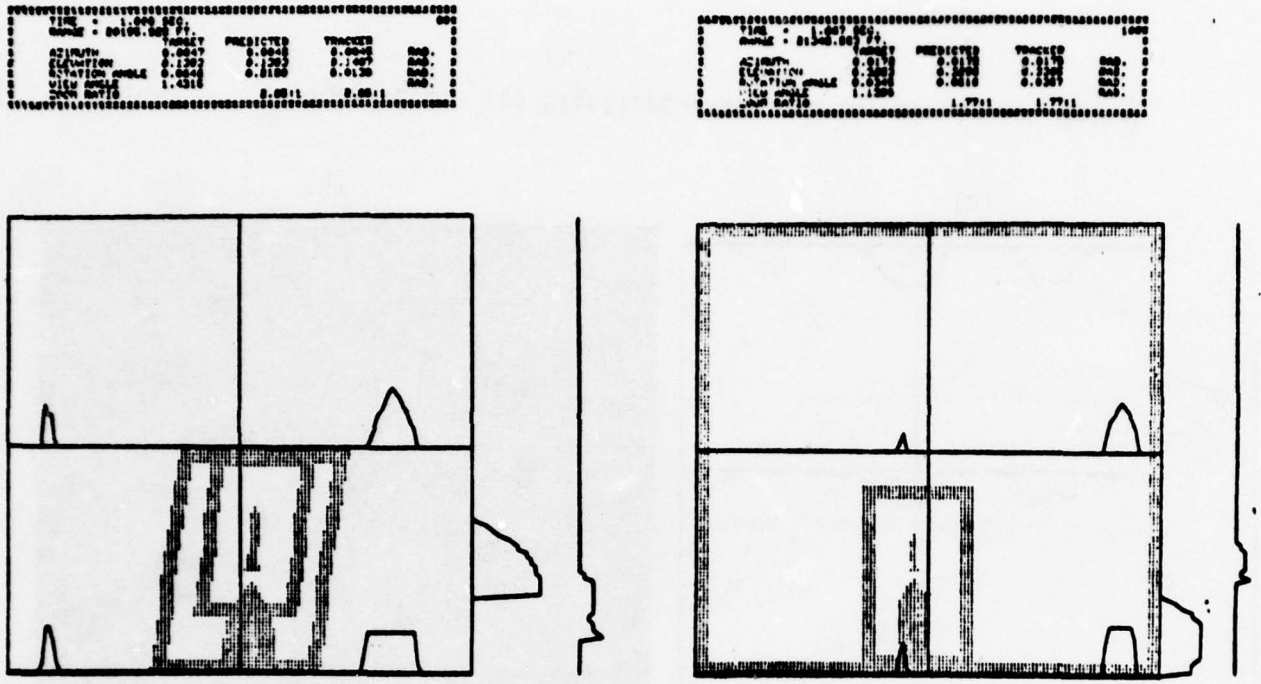
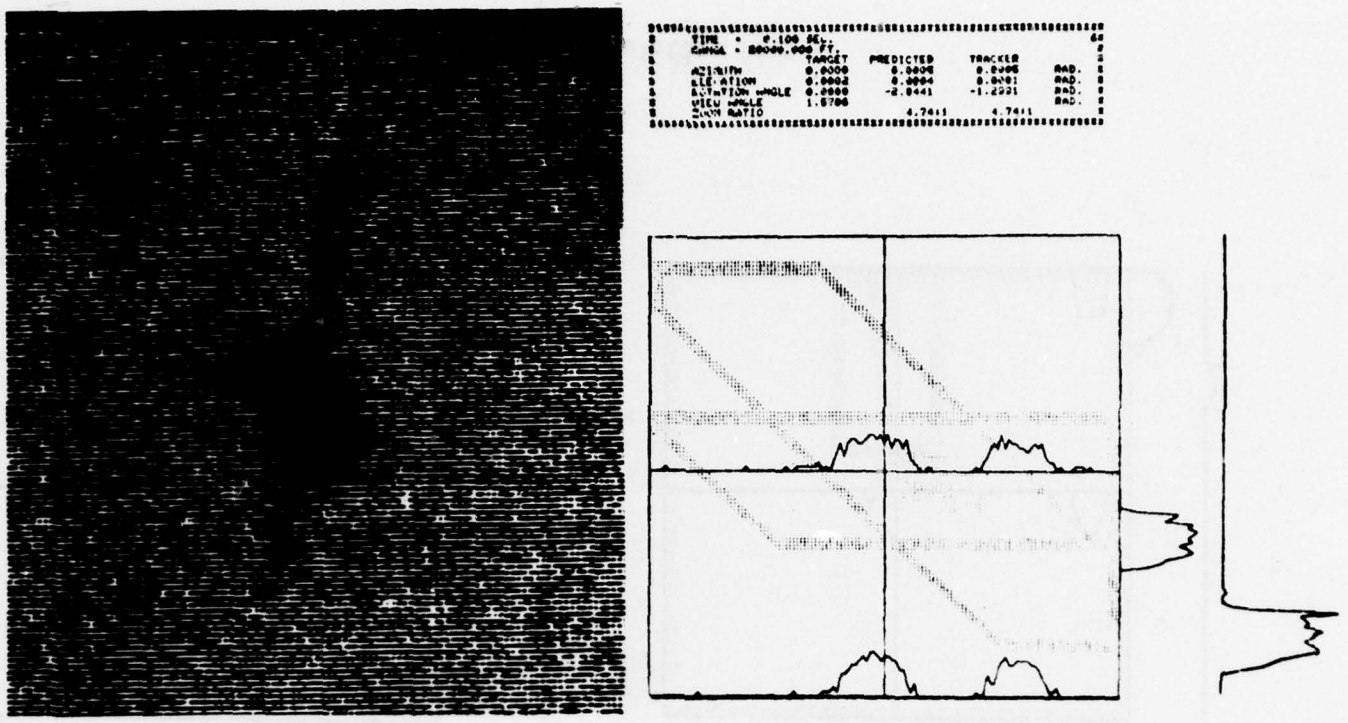


Figure 2.4. Simulation Outputs



(a) (b)

Figure 2.5. (a) Digitized Video and (b) Simulation Output

SECTION 3. TOOLS DEVELOPED AT WSMR FOR OPTICAL TRACKING RESEARCH

THE IMAGE PROCESSING LABORATORY

An image processing laboratory has been developed by the Advanced Technology Office, Instrumentation Directorate, WSMR, to support ongoing real-time optical tracking research. The image processing system is built around a PDP 11/35 minicomputer with two floppy disk drives, a Tektronix 4014 graphics display terminal, and a Tektronix 4631 hardcopy unit. Peripheral support equipment includes a Sony VO-2850 U-Matic Videocassette recorder, two Sony TV monitors, an IPS video disk, a Bright 7-track write-only magnetic tape drive, several TV cameras, a Consolidated Video Systems 504 digital video signal corrector, and a custom-built ISI digitizer. A Printronix line printer interfaced to a PDP LSI-11 is also available in an adjacent laboratory for displaying digitized picture files.

The present system configuration is shown in figure 3.1. A video tape containing a typical tracking sequence is played, or a TV camera is used to monitor a test mission. When an interesting tracking situation occurs, the video disk record button is pushed causing a sequence containing from one to 300 TV frames to be stored on the disk. Either video field of each TV frame stored on the video disk may be selected for digitization by the computer software which is input from a floppy disk. The digitizer converts the video field into a 512 x 240 (or 256 x 240), 8-bit pixel field. Thus, each of the 122,880 (or 61,440) pixels is quantized into one of 256 possible intensity levels, and this array of intensity values is stored on a floppy disk. The pixels are stored in 512 (or 256) columns with 240 pixels per column. The data is transposed to a row-structured file before the image is processed. The video signal corrector is used to produce the proper peak video amplitude and phase required by the video disk and the digitizer.

A library of 20 digitized images is presently available for processing at WSMR. These images contain targets and backgrounds which are representative of those to be encountered by the real-time video tracker. Figure 3.2 presents several examples from this library of images. A TV camera is also being mounted on a cinetheodolite at WSMR to obtain video sequences from the optical system on which the real-time video tracker will be deployed during fiscal year 1979. Sequential video frames from the cinetheodolite system will soon be added to the current library of images. This library has provided the basic image data needed to evaluate the effectiveness of many novel image processing algorithms including those to be deployed with the real-time video tracker as well as several promising new techniques presently being investigated by research personnel at WSMR. With the addition of sequential digitized frames of

video to the library of images, the entire target tracking loop can be evaluated using the real-time video tracker simulation at WSMR. Research personnel at various laboratories throughout the world will find this library of images to be a useful source of data to evaluate future intelligent optical tracking techniques.

The magnetic tape drive is used to transfer image files to 7-track tape for distributing the library of digitized images to interested research installations. Software has been developed at WSMR which allows these tapes to be formatted for compatibility with most computer systems. In addition, the data can also be transferred to 9-track tape or to computer cards if necessary.

The main software programs developed and being used in the WSMR image processing laboratory are listed in table 3.1 together with a brief description of the purpose of each program. These programs are used to digitize, transpose, trim, process, and display images; plot gray level histograms; and copy images to magnetic tape for distribution to other laboratories.

THE RTV SIMULATION AND THE RTV TRACKER AS RESEARCH TOOLS

The RTV simulation is being used as a research tool at WSMR. It is especially effective in evaluating the RTV system performance and in identifying and seeking solutions to real-time tracking problems before the RTV tracking system is deployed. With the added capability of using digitized video from a variety of tracking sequences as inputs to the video processor, the simulation can now test the system performance under a variety of tracking conditions, thus allowing thorough evaluation and possible refinement of the tracking and processing algorithms and the state transitions of the tracker processor.

When combined with the WSMR image processing laboratory, the simulation becomes a powerful aid in evaluating the performance of novel image filtering algorithms. Images can be digitized and processed using a variety of filtering techniques. The effectiveness of the filter algorithms can then be evaluated by comparing the simulation results obtained when original and processed image fields are used as inputs to the video processor.

The simulation can also be used to test new design concepts. Since it is implemented with a modular structure, the processing and tracking algorithms can be changed easily, allowing new algorithms to be evaluated within the framework of the complete tracking system. For example, features other than intensity could be derived from the relationships between sampled pixels in the three regions of the target tracking window. Possible candidates include texture, gradient, and linearity

TABLE 3.1 WSMR IMAGE PROCESSING SOFTWARE

Program	Language	Purpose
LPORIG LPSMAL LPLARG	MACRO MACRO MACRO	Gray level display of digitized picture data on a Printronix line printer
WRP256 WRP512	FORTRAN FORTRAN	Gray level display of digitized picture data on a Tektronix 4014 display terminal.
HISTP	FORTRAN	Histogram plot of the number of pixels at each of the 256 gray levels in an entire digitized image.
DHIST	FORTRAN	Histogram plot and gray level display of a specified portion of a digitized image.
TR256 TR512	FORTRAN FORTRAN	Digitized image transpose from column structured to row structured pixel storage.
TRIM	FORTRAN	Selection and storage of a specified portion of a digitized image.
DIG5F1 DIG5F2 DIG2F1 DIG2F2	MACRO MACRO MACRO MACRO	Digitization of field 1 or field 2 of a video frame into a 512 x 240 or 256 x 240 pixel field.
MAG256 MAG512	MACRO MACRO	Transfer of digitized image files from floppy disk to 7-track magnetic tape.
FILTER	FORTRAN	Selected filtering of digitized images. Filter types include average, median, human visual, sobel, maxmin, and moment.
COPY	FORTRAN	Copying and formatting of magnetic tapes for distribution.
PICTPR	FORTRAN	Target boundary extraction from image data.
STRUCT	FORTRAN	Structure analysis of target boundary.

measures, each of which could be implemented by modifying the image decomposition algorithm.

Once deployed, the RTV tracking system will become a versatile research tool with the tremendous advantage of real-time processing of video fields. The research and evaluation capabilities of the simulation are also present in the RTV tracker system, but with the added capability of obtaining immediate results. Each of the four processors will have a 1k x 48 writable control store for implementing the tracking algorithm, and the input/output processor will facilitate the reprogramming of the control stores with new algorithms. Additional control stores can be implemented on an extender card if future tracking algorithms require more than 1k of microcode.

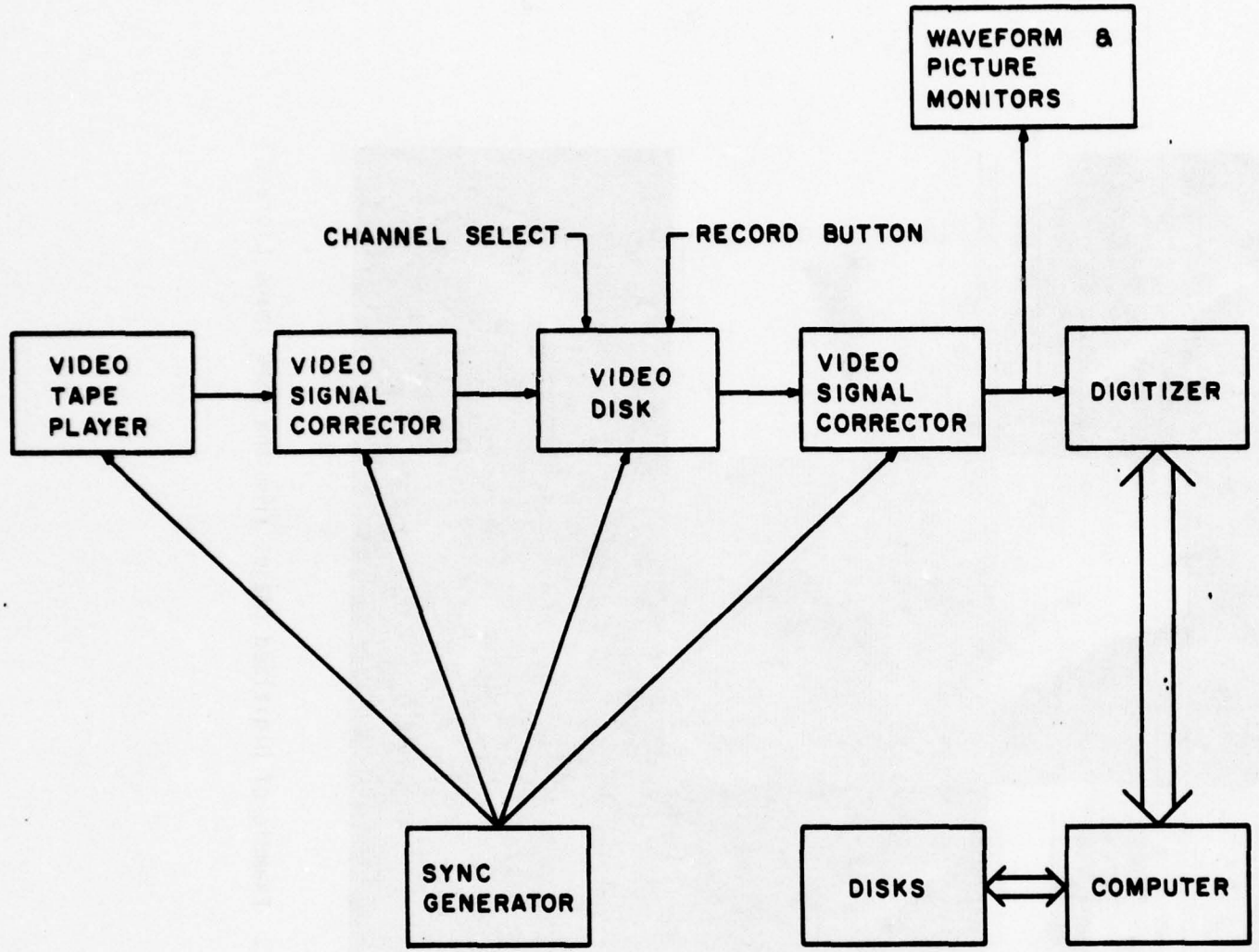


Figure 3.1. Video Digitization and Processing System

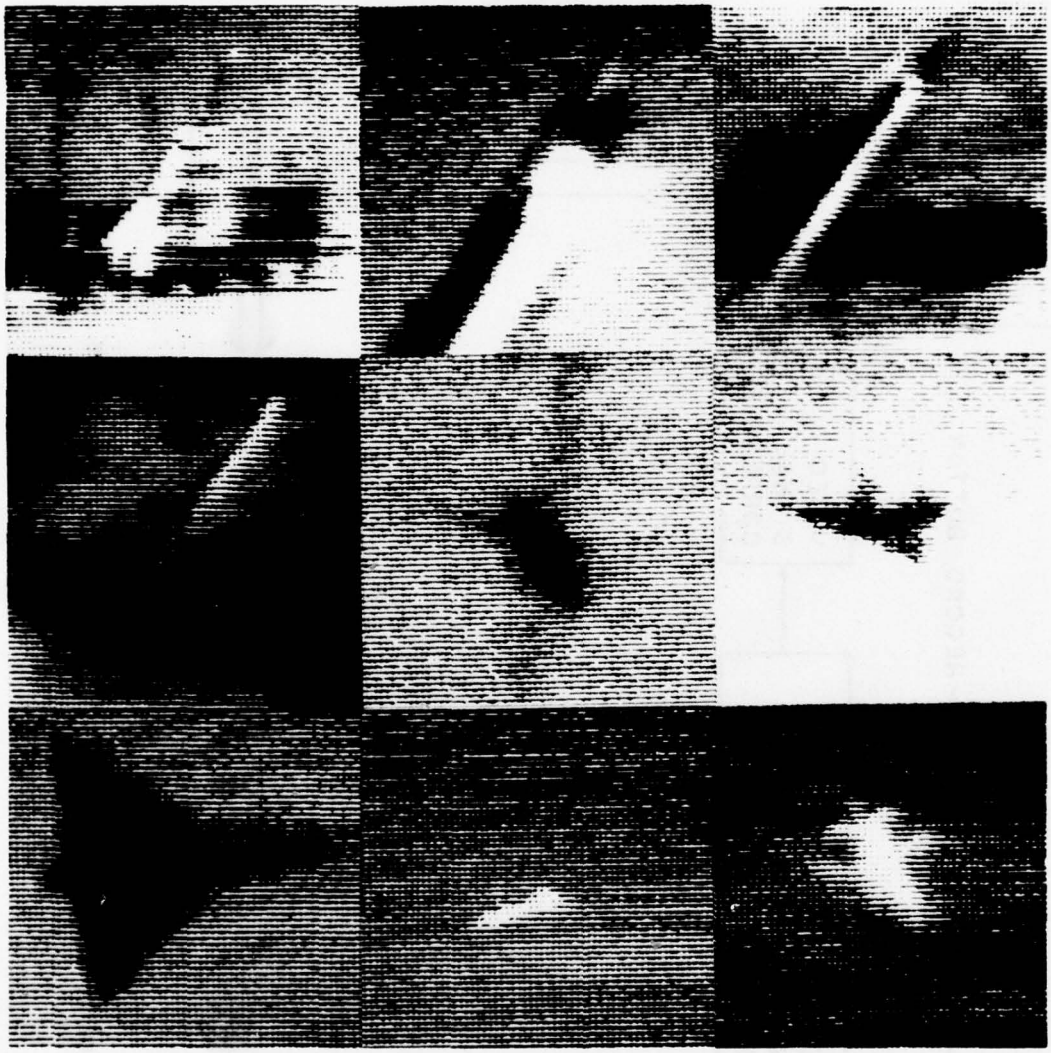


Figure 3.2. Example of Digitized Images from the WSMR Image Library

SECTION 4. RESULTS OF WSMR IMAGE PROCESSING RESEARCH

The development of the WSMR image processing laboratory has enabled WSMR personnel to investigate several novel image processing techniques in an effort to develop an RTV tracking system that is robust under the normal variety of tracking situations. The research can be divided into two major areas: (1) extraction of potential targets from the background, and (2) classification of the extracted objects as targets and nontargets. Research personnel at Purdue University, NMSU, and the University of Arizona's OSC have contributed to the success of the WSMR image processing effort.

THRESHOLDING TECHNIQUES FOR TARGET EXTRACTION

The simplest methods available for extracting potential targets from the background utilize thresholding operations. The method used by the video processor in the RTV tracker is a variation of thresholding in which the histogram of the background region of the tracking window (assuming a target with no plume) is used to identify potential target points. For example, if the intensity of a pixel within the target region does not occur within the background region, the pixel is identified as a potential target point. The threshold in this case is defined by the distribution of intensities in the background histogram.

A typical digitized video image which demonstrates the utility of this method is shown before and after target extraction in figure 4.1. Histograms of the target and background regions are included in figure 4.2. These gray level plots and the associated histograms were generated by allowing the DHIST program to operate on selected windows within the background and target regions of the digitized image. The target region in this case is the entire picture, while the background includes the regions near the top and bottom edges of the picture. Using DHIST, analyses of a variety of tracking images have verified the effectiveness of histogram comparison method of target extraction for most tracking imagery encountered at WSMR. The target intensities are easily identified in figure 4.2 as those intensities in the target region histogram which do not occur (or which occur less than about 30 times, in this example) in the background histogram; i.e., those intensities quantized at gray levels below 128 in the target region histogram. The gray level plot of figure 4.1(a) was generated by assigning a white level of 164 and a black level of 100 as the highest and lowest intensities, respectively, to be plotted as half-tone shades of gray. These limits include all intensity values which occur in the digitized image. The gray level plot of figure 4.1(b) was generated by assigning black and white levels of 100 and 128, respectively, thus causing all intensity levels having

a value of 128 or greater to be displayed as white. The half-tone pictures of figure 4.1 were generated using a 4 x 4 dot matrix for each resolution element; hence, 17 shades of gray are displayed.

Additional intelligence can be incorporated into the simple thresholding method of target extraction by using the measured histograms of target, background, and plume intensities to compute learned estimates of the probability density functions of intensities within each of the three regions of the tracking window. A Bayesian classifier can then use these estimates to extract potential target points. Additional features such as texture, gradient, and linearity might also be compared among sampled regions and used to devise an optimum extraction algorithm. Alternatively, the sampled regions could be reduced in size and increased in number to provide sampling windows over the entire image whose distinctive features are compared statistically to assign each window to a target, plume, or background region. Target extraction and region growing algorithms which incorporate these and other ideas are being developed for testing in the prototype RTV tracking system as part of ongoing research efforts at WSMR, NMSU, and Purdue University.

PREPROCESSING TO ENHANCE TARGET FEATURES

With the large variety of incoming images which are presented to an optical tracker, it may be necessary to process the images before target extraction is attempted to enhance distinctive features of the image which indicate the presence of targets. For example, man-made objects generally have distinctive textures and edges which may be enhanced and used in target extraction. The following preprocessing operations are being investigated at WSMR and Purdue University: a 2-dimensional median filter, an averaging filter, a 2-dimensional bandpass filter, a local extrema operator, and selected combinations of these operations. In addition, a maximum entropy image restoration technique (Frieden, 1977), developed at the University of Arizona's OSC, has been applied to WSMR tracking imagery. Other preprocessing operations to be investigated as part of the continuing research effort at WSMR include a variety of gradient operators, a moment operator, and a logarithmic input transformation.

All of the preprocessing algorithms currently being tested at WSMR are formulated in terms of filter windows which are either convolved with the digitized image or shifted and combined with the image pixels in a nonlinear fashion. The FILTER program allows the user to operate on any size input image by simply selecting the desired operation and window size. Presently only rectangular windows may be selected, and they must slide horizontally across the picture as the operation is performed.

Future research at Purdue University will investigate the use of a window whose shape is modified when edge points are detected, thus allowing nonrectangular windows to operate along target boundaries.

The median filter performs a nonlinear operation which eliminates high frequency noise while preserving monotonic edges precisely. A simple median filter replaces the intensity of the center pixel of each 3 x 3 window with the median intensity of all nine pixels within the window. Shown in figure 4.3 (upper left corner) is the original digitized image of an airplane turning away from the camera. The result obtained by processing this image with a 3 x 3 median filter window is shown in the upper right corner of figure 4.3. The edges are preserved, but the noise is reduced and the data is more correlated. The two bottom pictures of figure 4.3 illustrate the results obtained by applying the simple histogram comparison technique described above to display thresholded versions of the original (lower left) and median filtered (lower right) images.

The averaging filter replaces the central pixel of each window with the average intensity of all of the pixels within the window. The result is a smoothed version of the original picture. It is most effectively used in conjunction with the median filter. The median filter removes most of the noise points, and then the averager smears the remaining image points to produce an optimum image for thresholding. Figure 4.4 contains (from top to bottom) the original digitized image of a cruise missile, the median filtered image, and the result of averaging the median filtered image.

It is well known that the human visual system responds to incoming images with an approximate logarithmic transformation followed by a spatial and temporal bandpass filtering operation. The usefulness of such a spatial bandpass filter in extracting objects for tracking is being investigated.

A suitable filter function which is to be convolved with an input image is shown in table 4.1. The values are chosen to make computation of the fractional values easy for real-time implementation. Because the video data has twice as much resolution in the horizontal direction as in the vertical (only one video field is processed at a time), the bandpass filter is modified as shown in table 4.2. Both filters have a dc response of 0.5 and a gain of 7.5 for a small object which just covers the plus area.

The first two pictures in figure 4.5 present the results of convolving a digitized missile image with the filter function of table 4.2. The first picture is the original image. The missile becomes much more visible in the second picture because of the edge enhancement produced by the bandpass filter.

TABLE 4.1. 2-DIMENSIONAL BANDPASS FILTER FUNCTION

0	0	0	0	- .25	0	0	0	0
0	0	0	- .25	- .5	- .25	0	0	0
0	0	-.25	- .25	+ .25	- .25	-.25	0	0
0	-.25	-.25	+ .5	+1.0	+ .5	-.25	-.25	0
-.25	-.5	+.25	+1.0	+1.5	+1.0	+.25	-.5	-.25
0	-.25	-.25	+ .5	+1.0	+ .5	-.25	-.25	0
0	0	-.25	- .25	+ .25	- .25	-.25	0	0
0	0	0	- .25	- .5	- .25	0	0	0
0	0	0	0	- .25	0	0	0	0

TABLE 4.2. MODIFIED BANDPASS FILTER TO ALLOW HALF RESOLUTION IN VERTICAL DIRECTION

0	0	0	- .25	- .75	- .25	0	0	0
0	-.25	- .5	+ .25	+1.25	+ .25	-.5	-.25	0
-.25	-.5	+.25	+1.0	+1.5	+1.0	+.25	-.5	-.25
0	-.25	-.5	+ .25	+1.25	+ .25	-.5	-.25	0
0	0	0	- .25	- .75	- .25	0	0	0

The third picture (far right) in figure 4.5 shows the results of passing the digitized image through the bandpass filter and the local extrema operator. In a nxm window, the extrema operator compares the gray level of the center point with those of its two vertical neighbors. If it is above both neighbors, the center point is a local maximum in the vertical direction. If this is the case, the center point is compared with each point along each vertical direction until a gray level is encountered which is above the center point's value or until the edge of the window is encountered. The largest differences between gray levels in each vertical direction are then compared, and the smallest of the two is retained as the size of the local maximum in the vertical direction. An example is shown in table 4.3 for a 5 x 7 window.

The center value is 45. The 47 ends the search in the top direction, and the 50 ends the search in the bottom direction. The range is 15 above and 12 below. Therefore, the center point is a local maximum in the vertical direction of size 12. If the point is a local minimum instead of a maximum, the process is done in the same way, interchanging the above and below comparison tests.

This process is also done in the horizontal direction. In the example of table 4.3, the center point is not a local extrema in the horizontal direction. If a point is a local extrema in both horizontal and vertical directions, only the largest of the two is retained at that location. The extrema detection process is equivalent to local maximum and minimum determination following hysteresis smoothing of various amounts.

In figure 4.5 (right), the extrema sizes are indicated by the displayed gray levels. No distinction is made between horizontal and vertical extrema. The edges emphasized by the bandpass filter are marked by the extrema. The missile orientation can be extracted from its edge information.

The texture of various regions can also be characterized by the types and number of extrema present. The region characterization may be useful for background classification and for plume identification. Parameters for this measurement are extracted by counting the number of extrema of various sizes within a window surrounding each point.

The results of a maximum entropy restoration of the digitized image of a cruise missile are presented in the left half of figure 4.6. The bottom left picture is the original digitized image, the center picture is a smoothed image obtained by convolving the original image with a Gaussian spread function, and the top left picture is the restored image. By comparing the restored image with a scale drawing of the cruise missile (right half of figure 4.6), the remarkable accuracy of the restoration is verified, particularly in the wedge shape of the nose and the

TABLE 4.3. SAMPLE GRAY LEVELS FOR EXTREMA DETECTION

36	40	47	30	24
33	34	30	32	36
36	40	32	40	30
42	46	45	43	35
36	40	33	47	32
34	42	50	42	40
30	30	20	45	36

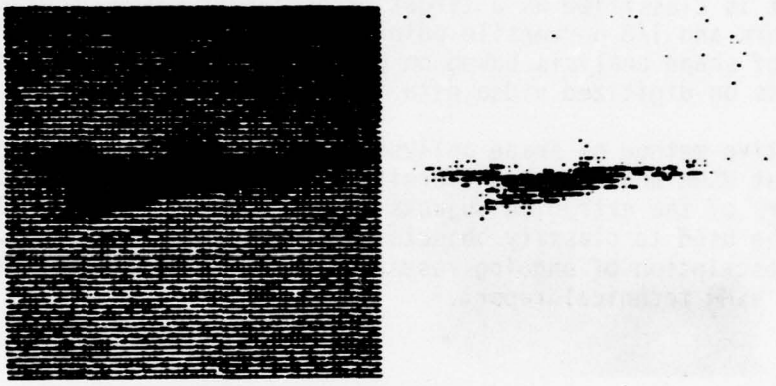
detail of the air inlet. The maximum entropy restoration cannot be accomplished in real-time. However, using array processors it may be possible to utilize this technique for near real-time applications which require the image to be restored within a few hundred milliseconds.

CLASSIFICATION OF EXTRACTED OBJECTS

As was described previously, the RTV tracker contains a projection processor which accumulates projections of the potential targets extracted by the video processor, and a tracker processor which utilizes the 1/8 percentile points of the projections to characterize the object shape. Each object is classified as a target or nontarget based on preassigned shape factors and 1/8 percentile points obtained from previous frames. This type of shape analysis based on projection data is producing excellent results on digitized video data in the RTV simulation.

An alternative method of shape analysis currently being investigated by personnel at WSMR and Purdue University utilizes Fourier descriptors of the contours of the extracted objects to generate complexity measures which can be used to classify objects as targets and nontargets. A detailed description of ongoing research in this area is presented in a previous WSMR technical report.[3]

3. Fukunaga, K., A. L. Gilbert, M. K. Giles, O. R. Mitchell, R. D. Short, and J. M. Taylor, "Segmentation and Structure Analysis for Real-Time Video Target Tracking," WSMR Technical Report, STEWS-ID-77-1, October 1977.



(a) (b)

Figure 4.1. Digitized Video Image (a) before Target Extraction and (b) after Target Extraction using a Thresholding Technique

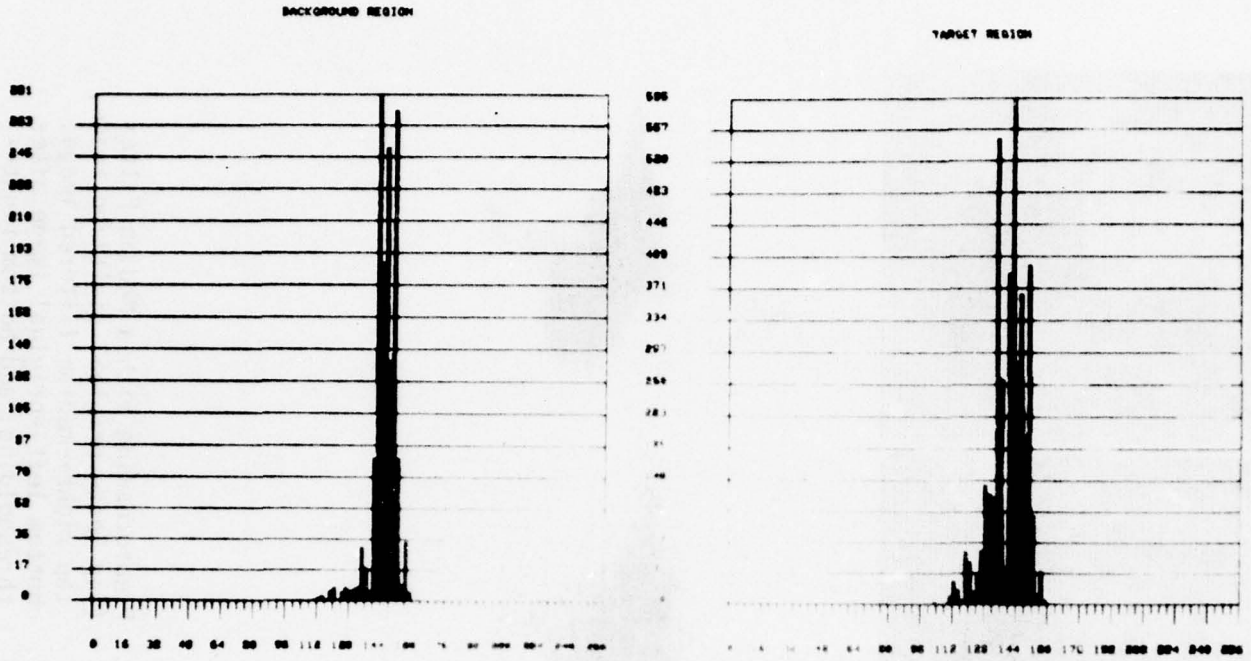


Figure 4.2. Intensity Histograms of Background and Target Regions of Figure 4.1(a)

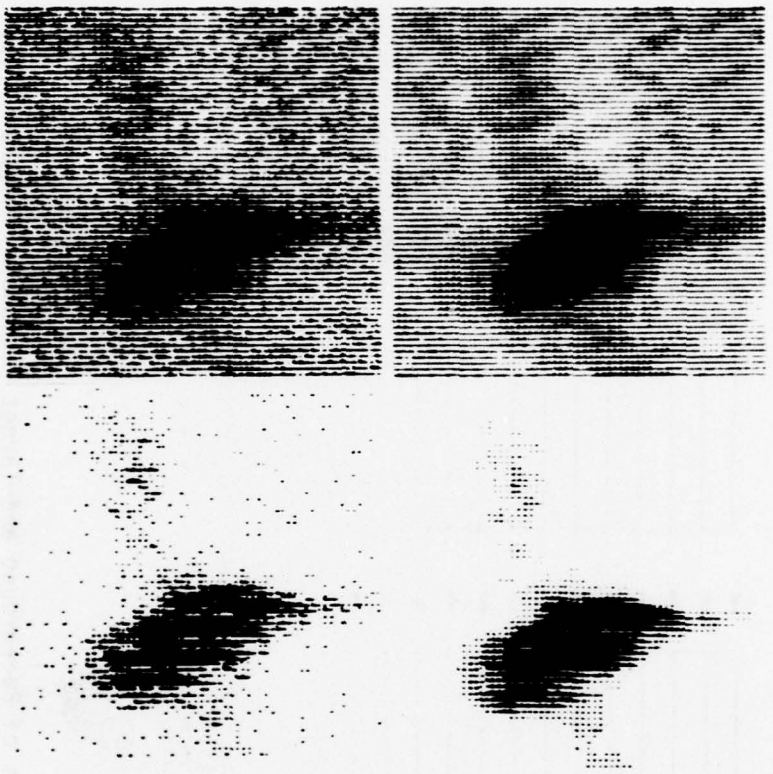


Figure 4.3. Preprocessing with a Median Filter (Top left--original digitized image, top right--median filtered image, bottom left--original image after thresholding, and bottom right--median filtered image after Thresholding)

Figure 4.4. Preprocessing with Median and Average Filters (Top--original digitized image, center--median filtered image, and bottom--median filtered image after averaging)

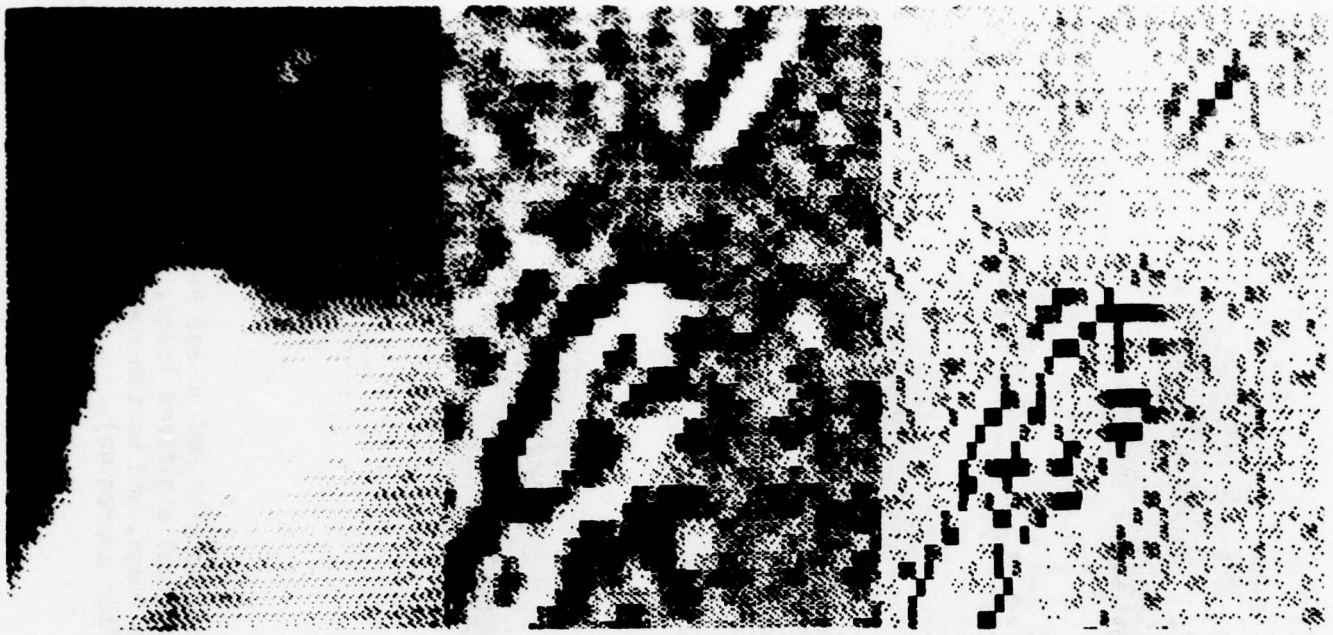


Figure 4.5. Preprocessing with Bandpass and Extrema Operators (Left--original digitized image, center--bandpass filtered image, and right--extrema detected in the bandpass filtered image)

Figure 4.6. Results of Maximum Entropy Restoration of a Cruise M (Bottom left is the original digitized image; center smoothed image; top left is the restored image. A scale drawing of the missile is shown on the right.)

APPENDIX A

SPIE INVITED PAPER

PATTERN RECOGNITION AND REAL-TIME BORESIGHT CORRECTION--A TUTORIAL

PATTERN RECOGNITION AND REAL-TIME BORESIGHT CORRECTION--A TUTORIAL (AN INVITED PAPER)

Dr. Alton L. Gilbert
US Army White Sands Missile Range
White Sands Missile Range, New Mexico 88002

Abstract

Advancements in technology over the past decade have opened doors for accomplishing computational tasks that were not imagined possible at the beginning of that period. Coupled with some recent concepts in pattern recognition and artificial intelligence, optical tracking system configurations with excellent tracking reliability and with the capability to correct for boresight error in real-time are within the scope of current technology. Aspects of the problem and the new supporting technology are discussed in this paper to put these developments into perspective.

Introduction

Optical tracking has been a mainstay of accurate metric range instrumentation since the first testing of modern rocketry during and following the Second World War. The accuracies that were possible from optical instruments exceeded those from other available instruments. Improvements in encoders, optical testing, modelling of the atmosphere, and optical design continuously improved the accuracies of optical instruments. The major drawback was the required film processing which delayed the delivery of boresight corrected optical data.

Recent changes in technology not directly related to optics have created the potential for relieving part of the delay problem in data delivery. Automatic tracking methods using high-speed microprocessors, artificial intelligence, and pattern recognition techniques, together with special modifications to the existing optical systems, are now available to perform most of the film reading function in an on-line, real-time mode. These methods far exceed the conventional contrast, edge, and correlation trackers in sophistication and capability, since they are based upon an understanding of some definable properties of the image involving many parameters as compared to only a few.

The Intelligence of Object Identification

Pattern recognition is a mathematical science based upon the separation of a parameter space into two or more regions, so that when the parameter

is measured it may be classified as belonging to one of the appropriate regions. It follows that a vector parameter will give rise to a parameter space of dimensionality equal to the number of independent elements in the vector implying that for an N-vector the required separation is a hyperplane in N space (See Figure 1). If the parameter is a single element vector, an assignment can be made on the basis of a single threshold on the real numbers and a tracker can be built that uses this decision rule. An example we call a contrast tracker uses a threshold on brightness for the assignment. A preprocessing algorithm may be placed before the decision. If we preprocess for magnitude of change in intensity, the same thresholding rule will yield an edge tracker. These are amongst the simplest applications of pattern recognition to the object identification problem. The very simplicity of the method produces the major drawbacks to the application. Since these algorithms are easily confused, many spurious objects in the field of view (FOV) often meet the classification criteria.

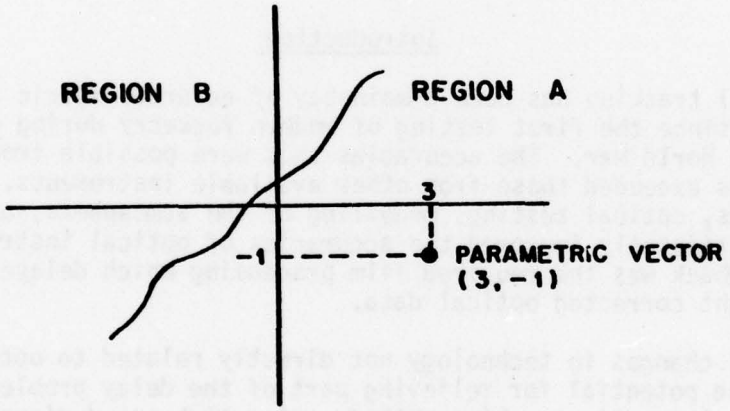


FIGURE 1

DECISION SPACE

A somewhat different approach that uses an array of points and measures the closeness of fit to a subsequently measured similar array, while choosing the best match as the correct location, is generally known as a correlation tracker. The decision is again based upon a single element parameter vector (the closeness of fit), but the preprocessing of data is much more elaborate. While this approach offers an improvement in confidence that

the correct object has been located if the object description is known, it suffers from two principal problems. The first is that generally the object being tracked changes appearance continually while objects in the background may not. This requires an adaptive object description which may slowly converge to the acceptance of an undesired object as the desired one. The second problem is that this approach requires a very large amount of processing to do a good job, since the optimal linear process would be convolution of the NxM object description array over the PxQ array of data points, and generally P>>N, Q>>M. A commonly used simplification is an additive (subtractive) algorithm that seeks the best fit of the desired array to the data, instead of the convolution. This approach necessarily results in loss of tracker performance. For these reasons, the correlation tracking method is generally limited to very restricted window tracking and fairly slow update rates.

Approaches to real-time optical tracking have generally been limited to these approaches, for the following principal reasons. The first and most important has been the magnitude of the real-time processing requirement for the more elaborate approaches, which have exceeded computational resources generally available. The second has been a lack of image understanding that would allow the formulation of more reliable, yet simple, approaches. Substantial progress has recently been made in the former, and there are many encouraging new developments in the latter.

A variety of methods of image data processing have become known over the past decade. Much of the effort has been sponsored by the Defense Advanced Research Projects Agency (DARPA) and by NASA to further scientific understanding of imagery and image processing. These agencies continue to sponsor image understanding research, and the various military services through their research sponsoring offices as well as the National Science Foundation have also become heavily involved in pattern recognition and image understanding research. Applications-oriented research at the US Army White Sands Missile Range (WSMR) and at the US Army Night Vision Laboratories (NVL) has lead recently to systems of reasonably high sophistication using concepts developed in-house and through sponsored research to solve complex identification and tracking problems. WSMR has concentrated on objects in the visible spectrum and in real-time, while NVL has been primarily concerned with the infrared and in near real-time. Many other systems, not necessarily real-time, have been developed for applications in medicine, meteorology, and space research.

Many of the newer methods involve the use of many elements in the parameter vector to glean more information from the data. In applying pattern recognition methods to the object identification problem, the engineer is trying to minimize the amount of data he must handle and maximize his confidence that he made the correct decision. Any linear process will preserve the quantity of data (260,000 points for a 512x512 image, possibly 8 bits per point) which is obviously not desirable if

much processing is required to make a decision. The engineer is forced to require a high degree of parallel processing on linear processes, and to perform nonlinear operations to reduce the data quantity prior to determining the values of the parameter elements used in the decision rule. Ideally, the dimensionality of the decision space should be kept reasonably small to allow decisions to be made in real-time or in near real-time.

Some of the preprocessing methods currently in use are:

Filtering

A general class of operations that involve the convolution of a point spread function array with the image to achieve some desired objective with the image. Examples include removing spatially invariant degradations due to the optics or atmosphere, boosting the high frequency content of the image to enhance edges, removing noise in the image, making the image more pleasing to the eye, and other such operations. Generally those operations which remove degradations are called estimation and those which emphasize some spatial frequencies or some aspects of the image are called enhancement. It must be noted that enhancement is an intentionally introduced distortion to produce some desired effect.

Transforms

The class of operations that maps the image into a new domain where the elements in the new domain are a measure of some property of the original image. The most common example is the discrete Fourier transform (DFT), especially in the fast algorithms (FFT). The DFT identifies the spatial frequency content of the image, which will allow further processing based upon these components. A class of binary Fourier (BIFORE) transforms has been developed over the past decade which are similar to the DFT but are more suited to computer applications. These might be called lesser transforms since they do not represent the information in the image as completely. Because they are much more efficiently run on a computer than the DFT, they have important applications in image transformations. Among the lesser transforms are the now popular Hadamard transform based upon Walsh functions, and the less known but simple Haar transform. To conceptually accommodate these lesser transforms, the notion of sequency as a conceptual equivalent to frequency was developed. Sequency is defined simply as half the number of zero crossings per unit interval. These transforms compute the concentration of image energy in these sequency components. This can be useful for identifying features of interest in the image. It is necessary, of course, to apply all of these transfroms in a 2-dimensional algorithm to process the 2-dimensional images. The Hadamard transform is particularly suited to computer operations on images.

Point Processing

Individual points in the image are assigned new values based upon some assignment rule. This may take a variety of forms with a large variation in apparent results. One point processing algorithm averages the corresponding point of several frames or sequential images to produce a weighted composite and remove transient degradations. Another assigns all values above a given threshold to 1 and all values below to 0. This is known as thresholding. A variation on thresholding is to assign predetermined gray levels to 1 even though these may not be in a continuous range. Still another algorithm, known as contrast stretching, assigns all values below some intensity I_0 to 0; all values above another intensity I_1 to the maximum gray level, say 256; and stretches the intermediate values to occupy the full range (see Figure 2). Generally, point processing methods are nonlinear, yielding fewer bits in the output than in the data array.

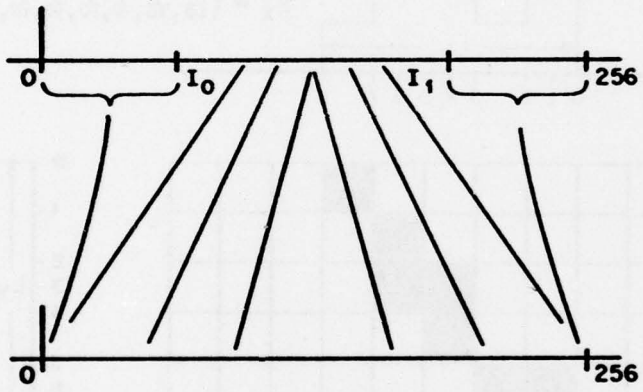


FIGURE 2

CONTRAST STRETCHING

The next step in the process is to identify the values of the elements in the parameter vector. These elements may be such things as size, orientation, number of corners, brightness, etc. When joined in a single parameter vector, they describe all we think we need to know to adequately describe the object for purposes of identification.

An example of this process of evaluating the elements of a parameter vector arises from the WSMR/New Mexico State University (NMSU) effort on the Real-Time Videotheodolite. The technique is creditable to Dr. Gerald Flachs and Yee HSun U of NMSU.

First, the video is point processed with an adaptive nonlinear algorithm that assigns each point either a 1 or a 0 depending upon whether, according to a Bayesian decision rule, it is classified as a potential target or probable background point. This yields one or more connected regions of 1's which then must be classified. A projection technique is applied for each such connected region where the number of 1's counted both horizontally and vertically are projected on the vertical and horizontal axes, respectively (see Figure 3). The projections are each segmented into eight equal area segments, and the segment lengths normalized by the projection length. The result is a parameter vector of 16 elements, eight horizontal projection segment lengths and eight vertical projection segment lengths. If the parameter vector is close to the stored

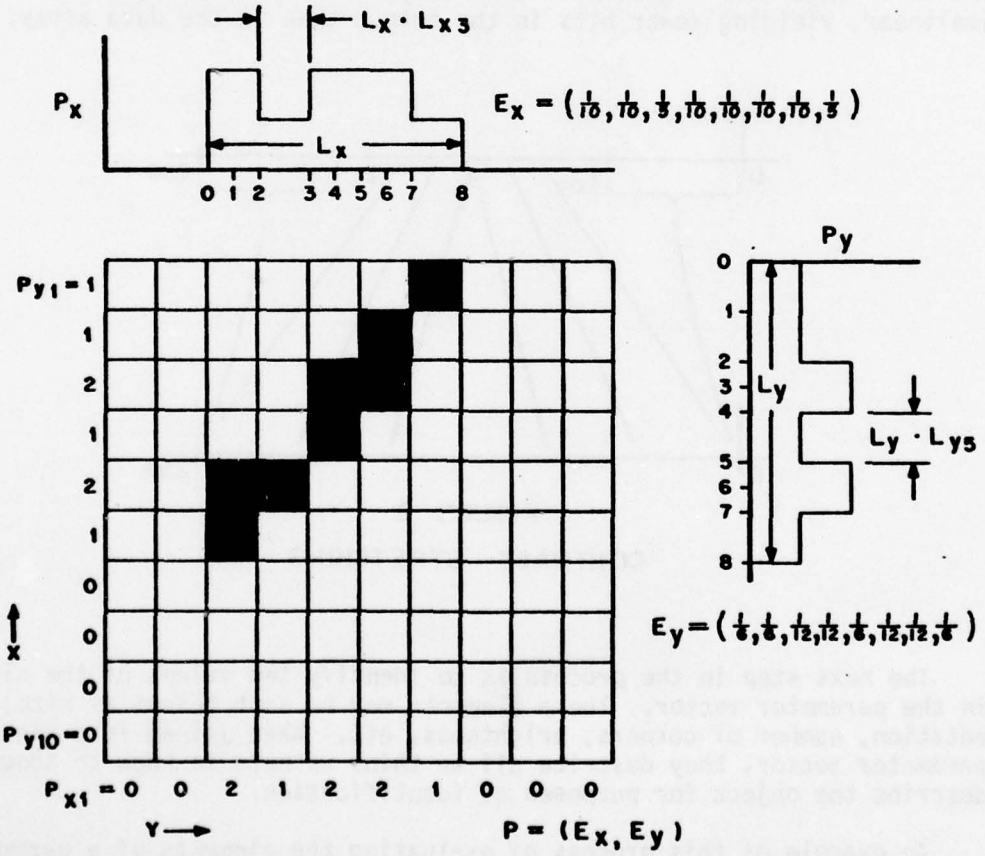


FIGURE 3
PROJECTIONS AND PERCENTILE POINTS

ideal vector in 16-space, a decision is made that the object being tested is the correct object for tracking. The actual WSMR/NMSU algorithm takes this one step further and separates the vertical projections onto the horizontal axis into a top and a bottom half to determine an orientation angle as well.

It can be seen that by using this method, the amount of data is rapidly reduced, first during point processing from 8 bits per picture element to 1 bit per picture element, and subsequently after the projections to only a few bits per image. This data compression is essential to enable currently available microprocessors to keep pace with the standard video rates of 60 fields per second, a goal accomplished by this system.

Finally, it should be noted that a human incorporates many elements into the parameter vector that he uses to identify an object. The difficulty of understanding the human visual process has caused rather slow progress in teaching computers to "see." We know that the human uses such things as texture, orientation, color, size, shading, shape, context, etc. to identify objects. Parameter vectors which incorporate these elements are difficult to compute, especially in real-time. Additionally, certain of these characteristics are difficult to quantify. It is not necessary, however, to require the computer to see the same things a human does. It is difficult to visualize elements of parameter vectors that do not have a physical meaning to a human, but which may be useful for computer recognition processes. Much work remains to be done to produce a highly sophisticated sight process in a computer.

Sensors and Associated Problems

The intelligence of object identification is only one of the obstacles to be overcome by the engineer. True, it has been one of the hardest problems to solve due to lack of understanding of image properties. Another troublesome problem, however, is the selection of a transducer on which to image the scene containing the object to be identified.

Generally there are three problems that must be overcome. These are resolution, timing, and sensitivity, each of which is discussed in turn. It is essential that these problems be isolated and solved to yield a system that will operate satisfactorily.

Resolution is an old problem well known to the optical engineer. It is simple to establish that resolution will not exceed the diffraction limit of the optical system. Generally, however, video transducers will not have near the resolution capability of the optical system to which they are attached. A good optical system may have 300 line pairs per millimeter resolution capability, with a vidicon attached that is limited to

about 40 line pairs per millimeter. This may not be sufficient for the required accuracy. A straightforward solution would be to interpose a magnification of five times so that the image plane resolution is 40 line pairs per millimeter with the transducer the same. This solution, however, forces much tighter control over the servo drives since the FOV is drastically reduced, and complicates the acquisition problem. A better solution is to use a zoom lens that may be driven to a wide FOV for high dynamic situations and acquisition, and a narrow FOV for low dynamic tracking. Design of the zoom lens will be complicated by focusing problems, run-out, required zoom transducers to adjust the magnitudes of the azimuth and elevation boresight corrections, and shuttering. Additionally, the zoom lens should not severely degrade the optical quality of the instrument to which it is attached.

It is noteworthy that high resolution vidicons are available, but the data bandwidth is generally too large to allow real-time processing. If a sensor produces more than about 500 measurable image points in 63.4 microseconds (standard TV) no present sequentially operated microprocessor can keep up. The high resolution vidicons, then, must be reserved for lower frame rates or larger computer processing arrangements.

Finally, video transducers are often subject to burns due to direct exposure to the sun. In these cases, it is necessary to provide a sun shutter or some means of protecting the device from burns.

Timing problems arise in a rather oblique fashion. It is no problem to synchronize video with IRIG A or B, apparently solving the timing problem. This is not true, however. Video sensors are generally scanned devices. The time elapse between the scanning of a single corresponding point one frame apart is well-established at 1/30 second for monochrome TV. For noncorresponding points, say (x_0, y_0) in frame 1 and (x_1, y_1) in frame 2, the time separation is not exactly one frame apart. Further, interlacing complicates the problem even more, since adjacent lines are separated in time by approximately 1/60 second.

For targets of high dynamics internal to the FOV, it is necessary to treat each field as a separate data sample array, and process it independently. This yields 60 samples per second which must be processed. The results may be used at that rate or averaged down to the more standard 20 samples per second range data rates.

The final difficulty is that even though adjacent lines are separated by 1/60 second, the object being tracked moves about in the FOV, and it is nonuniformly sampled in time. Of particular concern is variation in elevation since the scan process takes 1/60 second from top to bottom. The algorithms for estimating position at any given time, given a nonuniformly sampled data output, is rather complex, so another solution is desired.

The most straightforward is to gate the image onto the image plane, using a short enough gate to freeze motion. The gate may be spaced uniformly in time, thereby eliminating the nonuniform sampling problem. The image stored on the vidicon then may be scanned in the conventional manner, but the whole image may be treated as having occurred at some known time.

Finally, the question of transducer sensitivity must be considered. If it were not for the problems of resolution and timing, the sensitivity question would not arise. Generally, black and white vidicons are more sensitive than film (higher equivalent ASA). Addition of a 5X zoom will reduce available light by about five stops, but will vary throughout the zoom range. Additionally, if the vidicon is gated with, say, 50 microsecond windows at 60 gates per second, another 8 1/2 stops of intensity reduction is introduced. At this point, the transducer sensitivity will be inadequate for the T number of the system. It will generally be necessary to solve this problem through either the optical design (probably the expensive solution) or electronically. A reasonable approach is the use of an image intensifier that can incorporate the gating electronically, eliminating the need for mechanical shuttering and iris control.

Data Handling and Storage

It takes but a few moments of calculations to show that if each line is segmented into 512 data points, and there are 480 visible lines in a frame, all occurring in 1/30 second, then 7.4 million data points must be handled each second for full-frame processing (restricted window processing increases tracking difficulties in acquisition and tracking under high-dynamic conditions, and reduces resolution by competing with the zoom system for restricted FOV). Taking the data points at the rate they occur in the scanned line, each point must be completely processed in about 100 nanoseconds. For even the simplest point processing, these data rates represent a challenge. At present only a limited number of microprocessors are capable of approaching these speeds. These include the TI74S481 bit slice and the ADM29 family. The Real-Time Videotheodolite developed by WSMR/NMSU uses the TI74S481, which has been found to perform satisfactorily at a 200-nanosecond cycle time.

By using a nonlinear point process algorithm to compress the data and hard-wiring the algorithm in some fashion, the amount of data that results is within the capability of the microprocessor. Several special design techniques are necessary to handle and transfer the data from one processor to the next so that the whole tracking problem can be solved in real-time. The WSMR/NMSU effort uses multiple high-speed processors, each dedicated to a single task, to accomplish this objective.

Another problem is data storage. Since an event may occur (unexpected breaking-up of a missile, etc.) which may make the actual image very important in addition to the information normally being extracted, it is useful to store the video on a video tape recorder (VTR). To satisfactorily reconstruct the tracking problem such information as timing markers, shaft angle encoder information, equipment status, zoom and rotation drive settings, etc. is needed in addition to the video. This information is best stored together with the video on the VTR. WSMR has solved this problem by developing an interfacing data inserter that inserts this information into several lines of video available in the vertical retrace interval. This information is then stored with the video in a fashion that is readily recoverable if needed. A whole mission as viewed by the instrument may be reconstructed from one such video tape.

Conclusions

New methods to increase the utility of optical systems are being developed at a very fast rate. Other novel developments in technology that bear careful watching are optical computers, bragg cell technology using bulk acoustic wave devices, and on-chip charge-coupled device processing of visual information. A system that today pushes the state-of-the-art will be made obsolete by tomorrow's technology. There is probably more future in optical systems than in the purely electronic tracking systems of typical range instrumentation. Ideas such as those discussed in this paper are but stepping stones in the direction of truly intelligent, automatic optical tracking systems. The optical engineer has a challenge before him of staying up with a rapidly changing field and using new and powerful technology in the solution of optical tracking problems.

APPENDIX B

ARMY SCIENCE CONFERENCE PAPER

NOVEL CONCEPTS IN REAL-TIME OPTICAL TRACKING

GILBERT & *GILES

TITLE: Novel Concepts in Real-Time Optical Tracking

ALTON L. GILBERT and MICHAEL K. GILES
US Army White Sands Missile Range
White Sands Missile Range, New Mexico 88002

ABSTRACT:

Advancements in technology over the past decade have opened doors for accomplishing computational tasks that were not imagined possible at the beginning of that period. Coupled with some recent concepts in pattern recognition and artificial intelligence, optical tracking system configurations with excellent tracking reliability and with the capability to correct for boresight error in real-time are within the scope of current technology. Some aspects of the real-time optical tracking problem are discussed in this paper. A solution is described which incorporates image processing and pattern recognition techniques into a real-time optical tracking system. This system is under development at the US Army White Sands Missile Range.

BIOGRAPHY:

PRESENT ASSIGNMENT: Research Electronics Engineer, US Army White Sands Missile Range

PAST EXPERIENCE: US Navy, 1961-1968; Engineering Aide, New Mexico State University, 1968-1970; Research Engineer, New Mexico State University, 1970-1973; Associate Professor (Adjunct), New Mexico State University, 1973-1978; Research Electronics Engineer, US Army White Sands Missile Range, 1973-present.

DEGREES HELD: Bachelor of Science in Electrical Engineering (high honors), New Mexico State University, 1970; Master of Science, New Mexico State University, 1971; Doctor of Science, New Mexico State University, 1973.

NOVEL CONCEPTS IN REAL-TIME OPTICAL TRACKING

ALTON L. GILBERT, DR.
*MICHAEL K. GILES, DR.
US Army White Sands Missile Range
White Sands Missile Range, New Mexico 88002

INTRODUCTION: Optical tracking has been a mainstay of accurate metric range instrumentation since the first testing of modern rocketry. The accuracies that were possible from optical instruments exceeded those from other available instruments. Improvements in encoders, optical testing, modelling of the atmosphere, and optical design continuously improved the accuracies of optical instruments. The major drawback is the required film processing which delayed the delivery of boresight corrected optical data.

Recent changes in technology have created the potential for relieving part of the delay in data delivery. Automatic tracking methods using high-speed microprocessors, artificial intelligence, and pattern recognition techniques, together with special modifications to the existing optical systems, are now available to perform most of the film reading function in an on-line, real-time mode. These methods far exceed the conventional contrast, edge, and correlation trackers in sophistication and capability, since they are based upon an understanding of some definable properties of the image involving many parameters as compared to only a few.

THE INTELLIGENCE OF OBJECT IDENTIFICATION: Pattern recognition is a mathematical science based upon the separation of a parameter space into two or more regions, so that when the parameter is measured it may be classified as belonging to one of the appropriate regions. It follows that a vector parameter will give rise to a parameter space of dimensionality equal to the number of independent elements in the vector. Thus, for an N-vector, the required separation is a hyperplane in N space. If the parameter is a single element vector, an assignment

can be made on the basis of a single threshold on the real numbers, and a tracker can be built that uses this decision rule. An example we call a contrast tracker uses a threshold on brightness for the assignment. A preprocessing algorithm may be placed before the decision. If we preprocess for the magnitude of change in intensity, the same thresholding rule will yield an edge tracker. These are amongst the simplest applications of pattern recognition to the object identification problem. Since these algorithms are easily confused, many spurious objects in the field of view (FOV) often meet the classification criteria.

A somewhat different approach that uses an array of points and measures the closeness of fit to a subsequently measured similar array, while choosing the best match as the correct location, is generally known as a correlation tracker. The decision is again based upon a single element parameter vector (the closeness of fit), but the preprocessing of data is much more elaborate. While this approach offers an improvement in confidence that the correct object has been located if the object description is known, it suffers from two principal problems. The first is that generally the object being tracked changes appearance continually while objects in the background may not. This requires an adaptive object description which may slowly converge to the acceptance of an undesired object as the desired one. The second problem is that this approach requires a very large amount of processing to do a good job, since the optimal linear process would be a convolution of the NxM object description array over the PxQ array of data points, and generally P>>N, Q>>M. A commonly used simplification is an additive (subtractive) algorithm that seeks the best fit of the desired array to the data, instead of the convolution. This approach necessarily results in loss of tracker performance. For these reasons, the correlation tracking method is generally limited to very restricted window tracking and fairly slow update rates.

Approaches to real-time optical tracking have generally been limited to these approaches for the following principal reasons. The first and most important has been the magnitude of the real-time processing requirement for the more elaborate approaches, which have exceeded computational resources generally available. The second has been a lack of image understanding that would allow the formulation of more reliable, yet simple, approaches. Substantial progress has recently been made in the former, and there are many encouraging new developments in the latter.

A variety of methods of image data processing have become known over the past decade. Applications-oriented research at the US Army White Sands Missile Range (WSMR) has lead recently to a system of reasonably high sophistication using concepts developed in-house and

through sponsored research to solve complex identification and tracking problems. WSMR has concentrated on objects in the visible spectrum and in real-time. Many other systems, not necessarily real-time, have been developed for applications in medicine, meteorology, and space research.

Many of the newer methods involve the use of many elements in the parameter vector to glean more information from the data. In applying pattern recognition methods to the object identification problem, the engineer is trying to minimize the amount of data he must handle and maximize his confidence that he made the correct decision. Any linear process will preserve the quantity of data (260,000 points for a 512x512 image, possibly 8 bits per point) which is obviously not desirable if much processing is required to make a decision. The engineer is forced to require a high degree of parallel processing on linear processes, and to perform nonlinear operations to reduce the data quantity prior to determining the values of the parameter elements used in the decision rule. Ideally, the dimensionality of the decision space should be kept reasonably small to allow decisions to be made in real-time or in near real-time.

Some of the preprocessing methods currently in use are:

Filtering: Filtering operations generally involve the convolution of a point spread function array with the image to achieve some desired objective with the image. Examples include removing spatially invariant degradations due to the optics of the atmosphere, boosting the high frequency content of the image to enhance edges, removing noise in the image, making the image more pleasing to the eye, and other such operations. Generally those operations which remove degradations are called estimation and those which emphasize certain spatial frequencies or certain aspects of the image are called enhancement. It must be noted that enhancement is an intentionally introduced distortion to produce some desired effect.

Transforms: Operations that map the image into a new domain are called transforms. The elements in the new domain are a measure of some property of the original image. The most common example is the discrete Fourier transform (DFT), especially in the fast algorithms (FFT). The DFT identifies the spatial frequency content of the image, which allows further processing based upon these components. A class of binary Fourier (BIFORE) transforms has been developed over the past decade which are similar to the DFT but are more suited to computer applications. These might be called lesser transforms since they do not represent the information in the image as completely. Because they are much more efficiently run on a computer than the DFT, they have

important applications in image transformations. Among these lesser transforms are the now popular Hadamard transform based upon Walsh functions, and the less known but simple Haar transform. These transforms can be useful for identifying features of interest in the image. It is necessary, of course, to apply all of these transforms in a two-dimensional algorithm to process the two-dimensional images.

Point Processing: In point processing, individual points in the image are assigned new values based upon some assignment rule. This may take a variety of forms with a large variation in apparent results. One point processing algorithm averages the corresponding point of several frames or sequential images to produce a weighted composite and remove transient degradations. Another assigns all values above a given threshold to 1 and all values below to 0. This is known as thresholding. A variation on thresholding is to assign predetermined gray levels to 1 even though these may not be in a continuous range. Still another algorithm, known as contrast stretching, assigns all values below some intensity I_0 to 0; all values above another intensity I_1 to the maximum gray level, say 256; and stretches the intermediate values to occupy the full range. Generally, point processing methods are nonlinear, yielding fewer bits in the output than in the data array.

The next step in the process is to identify the values of the elements in the parameter vector. These elements may include such things as size, orientation, number of corners, brightness, etc. When joined in a single parameter vector, they describe all we think we need to know to adequately describe the object for purposes of identification.

A REAL-TIME TRACKING SYSTEM: By using the above concepts together with high-speed microprocessors and special optics, a real-time tracking system may be devised that demonstrates a substantial advantage over the contrast, edge, and correlation trackers currently on the market. The greatest challenge is that of doing "intelligent" processing of video data at the extremely high data rates of standard TV.

The development of an intelligent real-time video (RTV) tracking system has been accomplished through the cooperative efforts of research and development personnel at WSMR, New Mexico State University (NMSU), and the Optical Sciences Center of the University of Arizona. The prototype RTV processor is being assembled at NMSU, the automatic zoom lens and image rotator at the University of Arizona, and the system interfaces at WSMR. The system components will be integrated and the system deployed early in fiscal year 1979 as an add-on modification to the Contraves Model F cinetheodolite at WSMR.

Figure 1 is a block diagram of the RTV tracking system which shows the RTV processor as the central element. The RTV processor

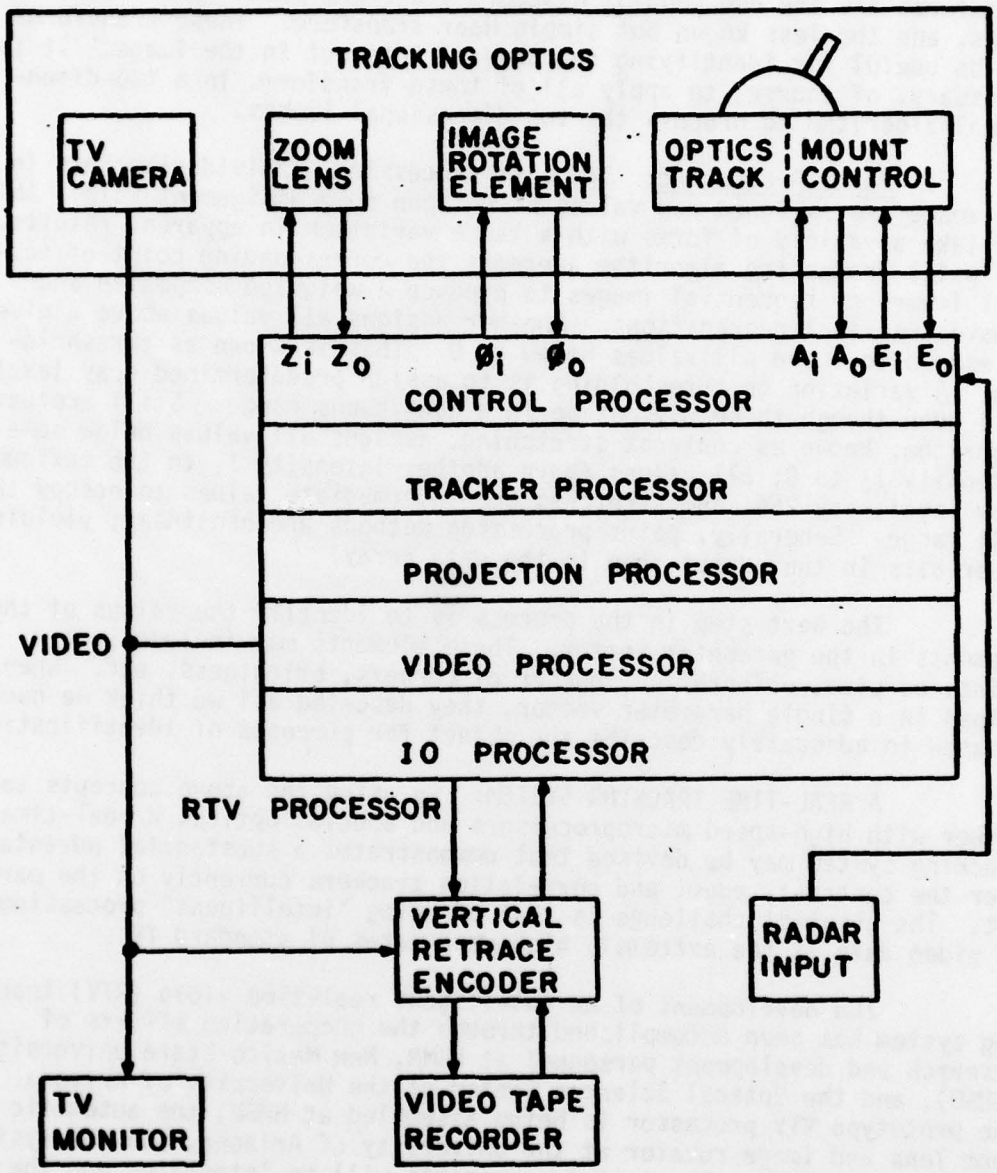


FIGURE 1. RTV TRACKING SYSTEM

receives standard composite video from a television camera, locates the target image, and provides control signals which drive the zoom and image rotation elements and point the Contraves tracking optics at the target. It also provides boresight correction signals and target attitude angles which are recorded into the vertical retrace period of the video tape used to record the tracking sequence.

The RTV processor consists of a distributive array of five processors, shown in Figure 1. The video processor synchronizes and digitizes the video signal from the TV camera, performs a statistical analysis of the digitized image, and separates the target images from the background. The projection processor accumulates binary projections of the target and plume images and establishes the structural parameters which locate and describe the shape of the target and plume images. The tracker processor establishes a structural confidence in the data and implements an intelligent tracking strategy. The control processor utilizes the structural confidence to combine current target coordinates with previous target coordinates to orient the optics toward the next expected target position, forming a fully automatic system. The input/output (I/O) processor provides a user interface to the tracking processors and is responsible for recording the tracking data with a video tape recorder.

A Research Oriented Processor Configuration: Four of the five distributive processors (excluding the I/O processor) which comprise the RTV processor are high-speed microprogrammable processors, each of which requires a stored microprogram to control its designated tracking function. To provide a powerful tool for future research in video tracking algorithms and to facilitate operational testing of the RTV system, the control store of each processor is realized with a read/write random access memory.

These four distributive processors are being built with a standard microprogrammable processor architecture to simplify the development and maintenance of the RTV tracking system. This standard architecture has been designed, built, and tested at NMSU. Based on the new Texas Instruments (TI) 74S481 Schottky processor chip, it provides a microinstruction cycle time of under 200 nanoseconds with sufficient computational power to implement the required RTV tracking algorithms. The standard architecture requires several LSI chips which may be partitioned into control and processing sections. Overlapping the execution of one microinstruction with the fetch of the next one allows the processor to achieve a minimum microinstruction cycle time equal to the larger of either the fetch time or the execution time, significantly increasing the speed of the processor.

The four high-speed processors included in the RTV tracking loop are described in some detail in the following paragraphs. In each case, the processor is built around the standard architecture outlined above. Some specialized hardware is added to the standard configuration in each case to accommodate the specific functions of the individual processors.

The Video Processor: The video processor decomposes each video field into target, plume, and background pixels at the standard video rate of 60 fields per second. As the TV camera scans the scene, the video intensity is digitized at m equally spaced points across each horizontal scan line. A resolution of m = 512 pixels per line results in a pixel rate of 96 nanoseconds per pixel. Within 96 nanoseconds, a pixel intensity is digitized and quantized into 8 bits (256 gray levels), counted into one of six 256-level histogram memories, and then converted by a decision memory to a 2-bit code indicating its classification (target, plume, or background). The 2-bit classification code is passed to the projection processor via the target data (TD) and projection data (PD) lines. TD is high for target points; PD is high for plume points.

The basic assumption of the image decomposition method is that the target image has some video intensities not contained in the immediate background. A tracking window is placed about the target image, as shown in Figure 2, to sample the background intensities immediately adjacent to the target image. The window frame is partitioned into two regions, B and P. Region B is used to provide a sample of the background intensities, and region P is used to sample the plume intensities when a plume is present. Using the sampled intensities, a very simple decision rule is used to classify the pixels in region T as follows:

- Background points--All pixels in region T with intensities found in region B are classified as background points.
- Plume points--All pixels in region T with intensities found in region P, but not found in region B, are classified as plume points.
- Target points--All pixels in region T with intensities not found in either region B or P are classified as target points.

A tracking window placed about the target image provides a method for sampling the pixel features associated with the target and background images. The background sample should be taken relatively close to the target image, and it must be of sufficient size to accurately characterize the background intensity distribution in the vicinity of the target. The tracking window also serves as a bandpass filter

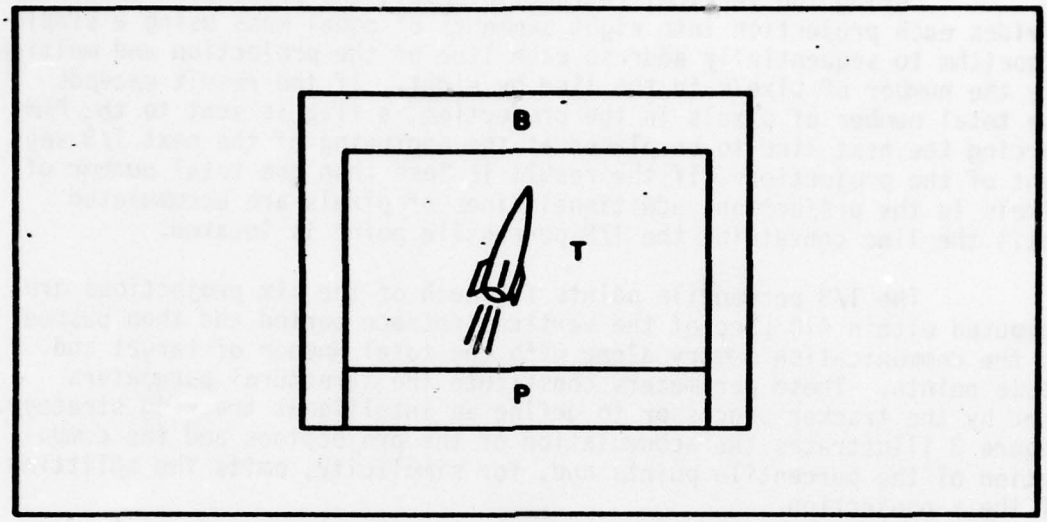


FIGURE 2. TRACKING WINDOW

by restricting the target search region to the immediate vicinity of the target. Although one tracking window is satisfactory for tracking missile targets with plumes, two windows provide additional reliability and flexibility for independently tracking a target and plume, or two targets. Having two independent windows allows each to be optimally configured and provides reliable tracking when either window can track.

The Projection Processor: The projection processor consists of a projection accumulation memory (PAM) and a standard processor which are designed to form projections of simultaneous target and plume windows and to compute structural parameters from the projections. The pixel data from each tracking window enters the PAM in real-time as a synchronized serial stream on lines TD and PD. As the classified pixel data is received, the PAM accumulates the projection data while the processor monitors the y-projections, accumulates the total number of target and plume points, and determines the midpoints used to split the x-projections. Each x-projection is split to allow the computation of target and plume attitude angles based on the locations of the median centers of the x- and y-projections of the top half and bottom half of the target and plume images.

During the vertical retrace interval, the projection processor divides each projection into eight segments of equal mass using a simple algorithm to sequentially address each line of the projection and multiply the number of pixels in the line by eight. If the result exceeds the total number of pixels in the projection, a flag is sent to the PAM forcing the next line to be placed at the beginning of the next 1/8 segment of the projection. If the result is less than the total number of pixels in the projection, additional lines of pixels are accumulated until the line containing the 1/8 percentile point is located.

The 1/8 percentile points for each of the six projections are computed within 410 μsec of the vertical retrace period and then passed to the communication memory along with the total number of target and plume points. These parameters constitute the structural parameters used by the tracker processor to define an intelligent tracking strategy. Figure 3 illustrates the accumulation of the projections and the computation of the percentile points and, for simplicity, omits the splitting of the x-projection.

Tracker Processor: The tracker processor receives the structural parameters from the projection processor, locates and characterizes the structure of the target and plume images, and decides on a tracking strategy to maintain track. It then outputs control signals to place the window frames in the video processor and outputs target location and orientation data to the control processor along with a confidence in the measured data. Since it operates on the projection data from field n while the projections for the next field (n+1) are being accumulated, the tracker processor is always one field behind the video and projection processors. The tracker and control processors must both finish their calculations before the vertical retrace interval begins for field n+1. This constraint requires the tracker processor to output its data to the control processor within 7 milliseconds after it receives the projection data.

Since the tracker processor is the only processor that communicates with all of the other three processors, each of which has its own coordinate system, the tracker processor must interpret the input data intelligently and then output the appropriate data to the video and control processors in their respective coordinate systems. The inputs are positive 16-bit integers defined for a coordinate system whose origin is the first pixel scanned inside the appropriate tracking window. The outputs to the video processor are 9-bit positive integers defined for a coordinate system whose origin is the first pixel scanned within the FOV. The 16-bit outputs to the control processor are defined for a coordinate system whose origin is the boresight.

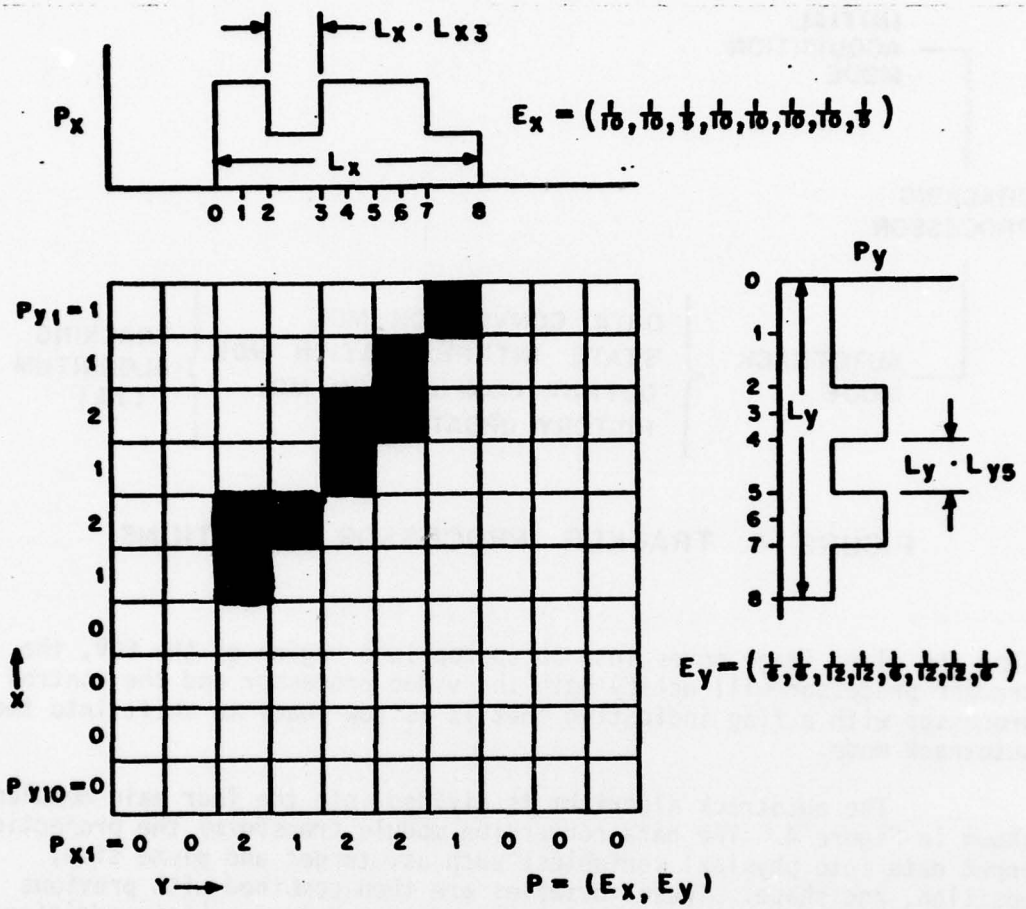


FIGURE 3

PROJECTIONS AND PERCENTILE POINTS

An overall view of the functions of the tracker processor is given in Figure 4. It has two modes of operation, the initial acquisition mode and the autotrack mode. The initial acquisition mode is used when the RTV system is trying to lock onto the target of interest. During this mode, the video processor does little or no learning on the target and plume intensities. The tracker processor will not instruct the control processor to begin predicting the target location until it is sure of the existence of at least the plume within the plume window.

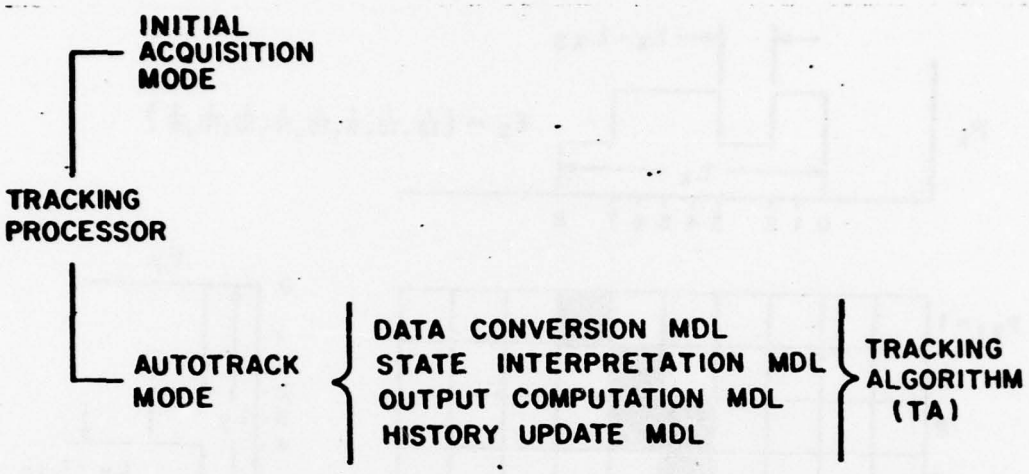


FIGURE 4. TRACKER PROCESSOR FUNCTIONS

When the plume image moves into an appropriate region of the FOV, the tracker processor will notify both the video processor and the control processor with a flag indicating that it is now ready to shift into the autotrack mode.

The autotrack algorithm is divided into the four main modules shown in Figure 4. The data conversion module transforms the projection input data into physical variables; such as, target and plume size, position, and shape. These variables are then combined with previous target activity data from the history update module to obtain additional variables; such as, the changes in target and plume position and size. All of these variables are compared with preassigned reference constants to obtain a set of binary inputs which are used directly by the state interpretation module to define the current tracking situation and produce an optimum tracking strategy. The strategy is implemented by the output computation module in the form of control signals to the video and control processors.

The Control Processor: The function of the control processor is to generate the four control signals that drive the real-time video tracker; i.e., the tracker azimuth A_i and elevation E_i which are sent to the RTV-Contraves system interface and the optics rotation ϕ_i and zoom Z_i which are sent to the RTV-zoom/rotation interface (Figure 1). In addition, the control processor outputs the following tracking data to

the I/O processor after each field so they can be recorded in the vertical retrace period of the video tape: field count, tracker status, time, x-displacement from boresight, y-displacement from boresight, tangent of the target orientation angle from vertical boresight, target azimuth, target elevation, tracker azimuth, tracker elevation, image rotation angle, and zoom ratio.

The tracking optics feeds the target image to the video processor portion of the RTV processor (Figure 1) which establishes the target coordinates with respect to the optics boresight. The control processor combines current target coordinates with previous target coordinates to point the optics toward the next expected target position. The predicted control equations are based on the combination of linear and quadratic optical estimates taken from a five-deep history stack. Since the input data is derived from field (K-1), and the estimates are being computed during field K, the control estimates must predict ahead two time increments to provide control signals which will place the boresight at the correct position during frame K+1.

COMPUTER SIMULATION OF THE REAL-TIME VIDEO TRACKER. A computer simulation of the RTV tracking system, incorporating the algorithms used in the control stores of the four distributive processors, has been developed and implemented on the PDP 11/35 system at WSMR. The purpose of this simulation is to provide a method for testing new design concepts and evaluating the RTV tracking system under realistic tracking conditions. The simulation model includes dynamic models for the target trajectory and the Contraves Model F cinetheodolite tracking system, in addition to the RTV processor algorithms, for simulating the complete tracking system. The recent development of an image processing laboratory at WSMR has enabled research personnel to digitize sequential video fields of typical tracking imagery. These fields of digitized video are now being used in the RTV simulation and in the development of improved image segmentation and structural analysis algorithms.

The RTV simulation is being used as a research tool at WSMR. It is especially effective in evaluating the RTV system performance and in identifying and seeking solutions to real-time tracking problems before the RTV tracking system is deployed. With the added capability of using digitized video from a variety of tracking sequences as inputs to the video processor, the simulation can now test the system performance under a variety of tracking conditions, thus allowing thorough evaluation and possible refinement of the tracking and processing algorithms and the state transitions of the tracker processor.

CONCLUSION: RTV tracking is not new, but recent developments have added new capabilities that enhance the advantages of these systems. Video tracking offers some distinct advantages over electronic

tracking (such as ECM immunity), but suffers from some disadvantages as well (such as restrictions in visibility). Several other aspects of system development for RTV tracking are discussed in the papers and reports listed in the bibliography.

A continuing research need exists for better understanding of imagery. A human incorporates many elements into the parameter vector that he uses to identify an object. The difficulty of understanding the human visual process has caused rather slow progress in teaching computers to "see." We know that the human uses such things as texture, orientation, color, size, shading, shape, context, etc. to identify objects. Parameter vectors which incorporate these elements are difficult to quantify. It is not necessary, however, to require the computer to see the same things a human does. It is difficult to visualize elements of parameter vectors that do not have a physical meaning to a human, but which may be useful for computer recognition processes. Much work remains to be done to produce a highly sophisticated sight process in a computer.

The concepts described in this paper have, however, been tested and will result in a prototype system deployed in 1979. Through a process of simulation and breadboard verification, WSMR has determined that such a system is well within the current capabilities of technology. A great deal of national (and some international) attention has been focused on this project because of the unique applications of pattern recognition in a tracking situation.

CHRONOLOGICAL BIBLIOGRAPHY: The papers and reports listed below have been produced during the course of this project.

1. Flachs, G. M., "A New Target Tracking Algorithm for the Automatic Programmable Film Reader," WSMR Technical Report, STEWS-ID-74-3, October 1974.

2. Flachs, G. M. and A. L. Gilbert, "Automatic Boresight Correction in Optical Tracking," Proceedings of the IEEE 1976 National Aerospace and Electronics Conference, June 1975.

3. Flachs, G. M., "Real-Time Tracking Algorithm," Final Report for Grant DAHCOA-15-G-0038, Army Research Office, October 1975.

4. U, Yee Hsun and G. M. Flachs, "Structural Feature Extraction by Projections," Proceedings of the IEEE Region V Conference, 1976.

5. Black, R. J., J. T. Whitney, G. M. Flachs, and A. L. Gilbert, "A Pre-Prototype Real-Time Video Tracking System," Proceedings of the IEEE 1976 National Aerospace and Electronics Conference, May 1976.

6. Flachs, G. M., W. E. Thompson, Yee Hsun U, and A. L. Gilbert, "A Real-Time Structural Tracking Algorithm," Proceedings of the IEEE 1976 National Aerospace and Electronics Conference, May 1976.

7. Thompson, W. E. and G. M. Flachs, "A Structure and Dynamic Mathematical Model of a Real-Time Video Tracking System," Proceedings of the IEEE 1976 National Aerospace and Electronics Conference, May 1976.

8. Flachs, G. M. and J. A. Vilela, "A Structural Model for Polygon Patterns," Proceedings of the IEEE 1976 National Aerospace and Electronics Conference, May 1976.

9. Gilbert, A. L. and G. M. Flachs, "A New Concept in Optical Tracking using Pattern Recognition," International Symposium on Information Theory, June 1976.

10. Vilela, J. A., "Application of Automata Theory to Picture Modeling," PhD Dissertation, New Mexico State University, 1976.

11. Whitney, J., "A Pre-Prototype Real-Time Tracking Filter," MS Thesis, New Mexico State University, 1976.

12. Flachs, G. M., W. E. Thompson, R. J. Black, J. M. Taylor, W. Cannon, R. B. Rogers, and Yee Hsun U, "A Pre-Prototype Real-Time Video Tracking System," Final Report for Contract DAAD07-76-C-0024, WSMR, January 1977.

13. Cannon, W., "A Microprocessor System Development Interface," MS Thesis, New Mexico State University, 1977.

14. Rogers, R. B., "Specification for a Two-Pass Symbolic Assembler Generating Absolute Microcode for Arbitrary Microprogram Control Store Architectures," MS Thesis, New Mexico State University, 1977.

15. Perez, P. I. and G. M. Flachs, "A Microprogrammable Processor Architecture for System Design," Technical Report, New Mexico State University, 1977.

16. Ritchell, J., "A General Purpose Microprogrammable Computer Architecture," MS Thesis, New Mexico State University, 1977.

17. Flachs, G. M., W. E. Thompson, J. M. Taylor, W. Cannon, R. B. Rogers, and Yee Hsun U, "An Automatic Video Tracking System," Proceedings of the 1977 National Aerospace and Electronics Conference, May 1977.

18. Rogers, R. B. and G. M. Flachs, "Mathematical Modeling and Simulation in a Programmed Design Methodology," Proceedings of the First International Conference on Mathematical Modeling, August 1977.

19. Fukunaga, K., A. L. Gilbert, M. K. Giles, O. R. Mitchell, R. D. Short, and J. M. Taylor, "Segmentation and Structure Analysis for Real-Time Video Target Tracking," WSMR Technical Report, STEWS-ID-77-1, October 1977.

20. Flachs, G. M., P. I. Perez, R. B. Rogers, S. J. Szymanski, J. M. Taylor, and Yee Hsun U, "A Real-Time Video Tracking System," Annual Report for Contract DAAD07-77-C-0046, WSMR, January 1978.

21. Gilbert, A. L., "Pattern Recognition and Real-Time Boresight Correction--A Tutorial," SPIE Proceedings, Vol. 134, Photo and Electro-Optics in Range Instrumentation, March 1978.

22. Giles, M. K. and A. L. Gilbert, "Concepts in Real-Time Video Tracking," WSMR Technical Report, STEWS-ID-78-1, May 1978.

23. U, Yee Hsun, "Applications of Projection Concepts to Image Processing," PhD Dissertation, New Mexico State University, 1978.

24. Gilbert, A. L. and M. K. Giles, "A Real-Time Video Tracking System," Proceedings of the 24th ISA International Instrumentation Symposium, May 1978.